Indiana University – Purdue University Fort Wayne
Opus: Research & Creativity at IPFW

Engineering Senior Design Projects

School of Engineering, Technology and Computer Science Design Projects

5-4-2011

A Wireless Energy Custodian Network

Nusaybah Abu-Mulaweh

Renee Chandler

Edwin Chobot

Daniel Newby

Indiana University – Purdue University Fort Wayne
Department of Engineering

ECE 406
Capstone Senior Design Project
Report #2

Project Title:	A Wireless Energy Custodian Network
Team Members:	Nusaybah Abu-Mulaweh Renee Chandler Edwin Chobot Daniel Newby
Faculty Advisor:	Chao Chen, Ph.D.
Consultant:	Carlos Pomalaza-Ráez, Ph.D.
Date:	5/4/2011

Table of Contents

Acknowledgment

We would like to express our sincerest appreciation to everyone who helped us to make this project a possibility. First and foremost, our gratitude is extended to our advisor Dr. Chao Chen and consultant Dr. Carlos Pomalaza-Ráez. Without Dr. Chen's skilled guidance this project could not have succeeded. We thank Dr. Ráez for his counsel, knowledge, and expertise in the area of Radio Frequency Communications. We are grateful to Indiana University Purdue University-Fort Wayne's Wireless Technology Center and to the Institute of Electrical and Electronics Engineers Standards Education Committee for their generosity with regards to funding our project

Abstract/Summary

Abstract
This document defines the determined qualities required for the IPFW Spring 2011 ECE406 Senior Design Project: A Wireless Energy Custodian Network. The defined qualities are based on the system requirements clarified in the initial project summary statement. The project summary statement is listed within the following passage as a reference.

Summary
The main goal of this project is to design, build, and test a wireless sensor and actuator network for monitoring the energy usage of AC devices in a home environment. Each node in the network reads the energy usage of an AC device connected to it and wirelessly reports the readings to a central server. The server displays the readings from these nodes through a user interface on a computer in real-time, in such a manner that users can understand their electricity usage patterns and adapt their behavior to reduce energy consumption. Moreover, users are able to remotely control the power On/Off of individual devices via the central server.

SECTION 1: DESIGN DESCRIPTION

1.1 SYSTEM OVERVIEW

In a world of rising energy costs and dwindling natural resources capable of producing energy, people and businesses are starting to look for better ways to help reduce their increasing electric bills. One way of reducing these costs is to monitor how much power is being consumed in real-time and from that data make informed decisions about how to manage the electrical devices being powered. A system which can give users an idea of how much power is being, has been, and might be, consumed will allow them to adjust their habits and lower the costs.

We plan to design a wireless energy custodian network which will monitor the power usage of alternating current devices/appliances in a home environment. The system will consist of two or more 'node' modules and a Central Server Module. The nodes will record data about the power consumption of the devices/appliances that are connected, and wirelessly transmit that data to the central server for processing. The server displays the readings from these nodes through a user interface in real-time. The goal of this project is to help users better understand how power is consumed in their devices and adapt their behavior to reduce their energy consumption.

Most electronic devices, even if turned off, will continue to draw power from a standard electrical outlet unless the device is manually unplugged. This power is called "*standby power*." While individual electronic products might now draw enough power while in standby power to be noticed, the average American family has almost forty devices constantly consuming power. The standby power consumption of these devices accounts for almost 10% of household electricity use. Because of this, our design will integrate an actuator into each measurement node that will automatically turn on and off the power supply to the AC devices remotely.

The remote on/off control can also be used in other manners to further reduce energy. For example, the air conditioner can be turned on and off remotely based on the inputs from temperature sensor readings and energy usage; the lights can be turned on and off remotely based on the inputs from motion sensor readings and energy usage; the water heater can be turned off before midnight and turned on before sunrise. The design of these appropriate control mechanisms, however, depends on the specific devices and the habits of individual users. Therefore, in our project, we specifically target the standby power to illustrate the feasibility and functionality of the on/off control. Furthermore, this on/off control enhances the wireless network from monitoring only to including the actuator part, which extends the capability of the whole system and makes our design different from other products on the market.

Figure 1 shows an application of this wireless energy monitoring system in a home scenario, where the measurement nodes are connected to major home appliances in different rooms, and the central server module displays the energy consumption of these appliances on a computer screen.

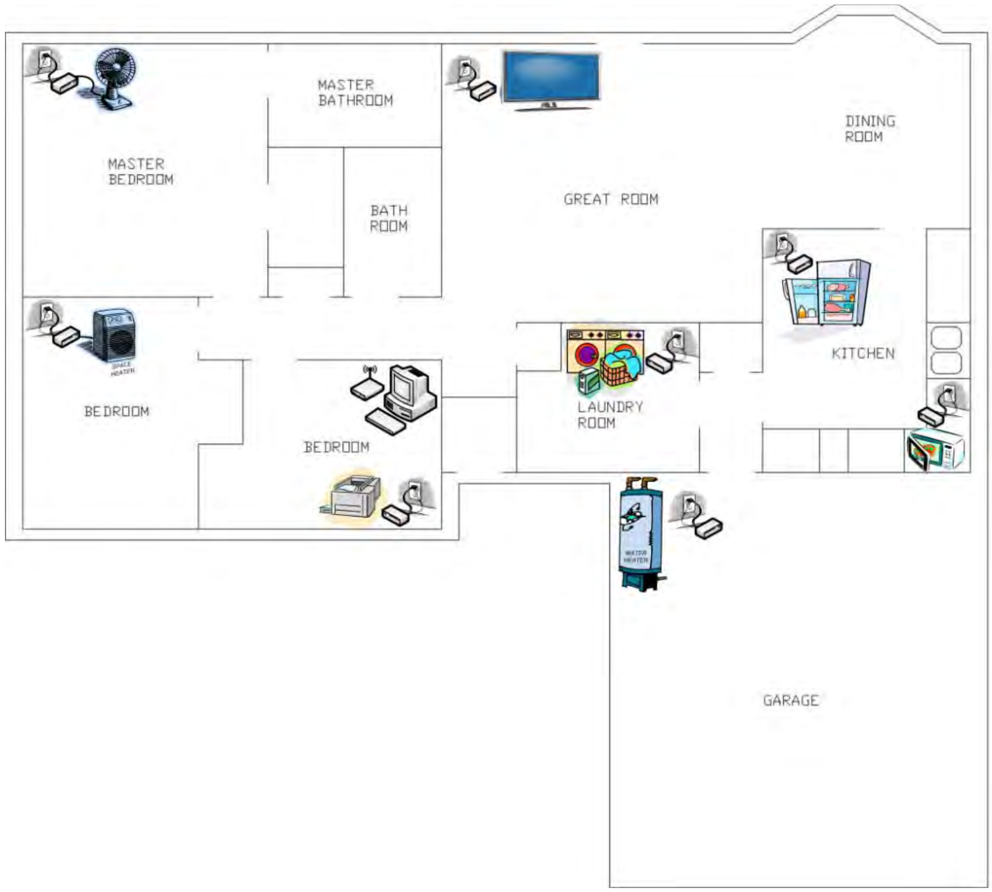


Figure 1: A wireless energy monitoring system at home.

Our group decided to design and implement a system prototype with two measurement nodes and one central server module, where the nodes communicate directly to the server and the server displays the measurement results through the computer. Other requirements of the system prototype include:

- The nodes monitor and wirelessly transmit the energy usage of connected AC devices.
- The central server module displays the energy reading in real-time through a graphical user interface and updating once every minute.
- The central server module is able to turn on and off the individual nodes. The on/off control will be tested in the application of standby power reduction.
- The system prototype will be deployed and tested indoor in a typical home environment in the United States. The communication distance from the measurement nodes to the central server module is within 15 meters.

Some products already on the market which function in a similar manner as the system we plan to design are the TED 1000 Series and Kill-A-Watt. The TED 1000 series takes the readings from an electrical panel in a home and tracks energy usage of an entire household and can display this information on a computer using the included software [8]. Kill-A-Watt is a device which monitors the amount of energy consumption of a connected appliance by the kilowatthour and displays it on an LCD display [6]. Our design is capable of controlling and monitoring multiple devices while neither the TED 1000 Series nor Kill-A-Watt can do this. Table 1 shows specific similarities and differences between these three systems.

Table 1: A summary of current energy monitoring systems on the market versus our design.

	TED 1000 Series	**Kill-A-Watt**	**Our Design**
Monitors	Yes	Individual Devices	Individual Devices
Real Time	Yes	Yes	Yes
Data Logging	Yes	X	Yes
Wireless Data Acquisition	Yes	X	Yes
Controls Connected Devices	X	X	Yes
Monitors Multiple Devices	X	X	Yes
Graphical User Interface	Yes	X	Yes
Export Data Capability	Yes	X	Yes

1.2 PROJECT DESIGN

Our system prototype has two measurement nodes and a central server module. Each measurement node will be plugged into a standard NEMA 15-5 electrical outlet. An AC device will then be plugged into the node for power measurement. Each measurement node contains the components necessary to measure the power consumption, wirelessly transmit the information to the central server module, and control the power on/off of the connected device. In order to measure the power consumption, a voltage divider network and low impedance current sense resistor are connected to an energy metering chip. The energy metering chip samples the voltage and current signals, calculates the power consumption, and outputs a pulsed digital signal with a frequency related to the power consumption. The microcontroller then samples the pulsed digital signal from the energy metering chip and converts to power. The microcontroller also integrates the power measurement samples to get the energy consumption and reports to the server through the wireless transceiver. The microcontroller also has the capability of controlling the power delivered to the connected device with a solid-state relay. An indicator LED will identify the node's current mode of operation, and a reset button will allow the user to restore power to a device that is currently off. The layout of the node is shown in Figure 2below.

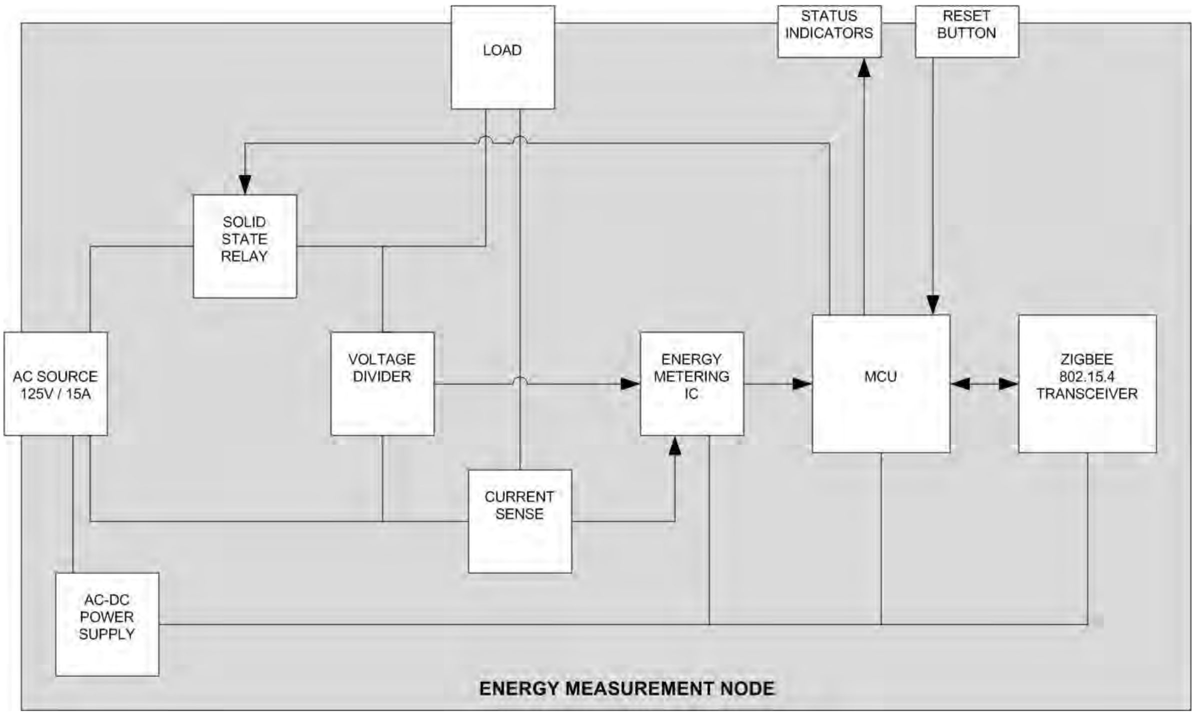


Figure 2: The block diagram of a measurement node.

The central server module will receive the power measurements from each of the nodes and forward the measurements to the computer for display. The measurement data is received through a ZigBee transceiver and passed directly to the computer program through the USB port. The power on/off signals from the computer program are passed to the ZigBee transceiver through the USB port. The components in the central server module are powered through the USB port as well. Figure 3shows a block diagram of the main components included in the central server module.

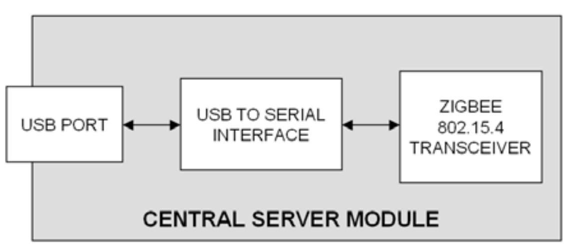


Figure 3: The block diagram of the central server module.

To summarize, the measurement nodes and the central server module have the following major components.

Measurement Node Components:

- *An AC-to-DC power supply* that will supply power to the energy monitoring circuitry.
- *A current to voltage conversion circuit* that will facilitate the measurement of the current flowing to the AC device.
- *An energy measurement circuit* that samples the continuous current and voltage signals in order to compute the energy consumption of the attached device.
- *A microcontroller* that performs the necessary calculations and then send the resulting data to a transceiver for transmission.
- *A wireless transceiver* that handles transmissions between the node and the server.
- *An actuator* that allows the user or the server to control the power on/off of the attached AC device.
- *Indicators* that alert the users of the operating status of a measurement node.

Central Server Module Components:

- *A wireless transceiver* for wireless communication with the measurement nodes.
- *A USB port* that can be used to connect to a personal computer.
- *A graphical user interface (GUI)* that allows the user to view the energy usage in real-time and track past power usage.

SECTION 2: BUILDING PROCESS AND CHANGES

2.1 OVERVIEW OF THE BUILDING PROCESS

In order to ensure that the proposed design would work as expected, our group performed several initial tests to several of the subsystems for the design. This included testing the DC power supply design, solid-state relay, optocoupler, and energy-metering IC chip. A complete discussion of these test results can be found in section 3. Once the initial testing was complete, the final revisions were made to the schematics for the CSM and measurement node. This allowed for the final selection of electrical components to be purchased and for the printed circuit board layout to be created. After the layout was completed, we were able to purchase the printed circuit boards and other components required for the assembly of the measurement nodes.

2.2 SCHEMATICS FOR CENTRAL SERVER MODULE (CSM) AND NODES

The initially proposed design for the Central Server Module included a PIC24 microcontroller to handle communications between a RS-232 serial connection and the Xbee wireless transceiver. Our design was later changed after learning more about the FTDI chip from Future Technology Devices International. This device converts the UART communication interface used by the Xbee transceiver to the USB interface used on most personal computers. By using the FTDI chip in our design we no longer required a microcontroller to handle communications between the transceiver and the computer using our CSM software. Furthermore, our group was able to purchase an Xbee Explorer Dongle from Sparkfun Electronics [10] and use this device as the interface between the transceiver and computer. The Xbee Explorer Dongle offers the same functionality as our revised design for the central server module. The original and revised schematics for the CSM are shown below in Figure 4 and Figure 5, respectively.

The original proposed design for the measurement node also included a PIC24 microcontroller. This was changed to the Silicon Labs C8051F350 microcontroller [3] for several reasons. The first reason for this change was that our group already had access to a Silicon Labs development board for testing and software development. This eliminated the need to purchase a development board for the PIC24 microcontroller. Another reason for the change was the overall group knowledge with each microcontroller. All four members of our group have some experience with the 8051 microcontroller whereas only two members have experience with the PIC24 microcontroller. A final reason for choosing the Silicon Labs microcontroller over the PIC24 microcontroller was the component package options. The Silicon Labs C8051F350 is in a 32-pin QFP package and the PIC24 is in a 100-pin QFP package. The 32-pin package still offered all of the I/O connectivity required for our design and also made the PCB layout and assembly processes much simpler, thus improving the overall design. The schematic for the revised measurement node design is also shown below in Figure 6.

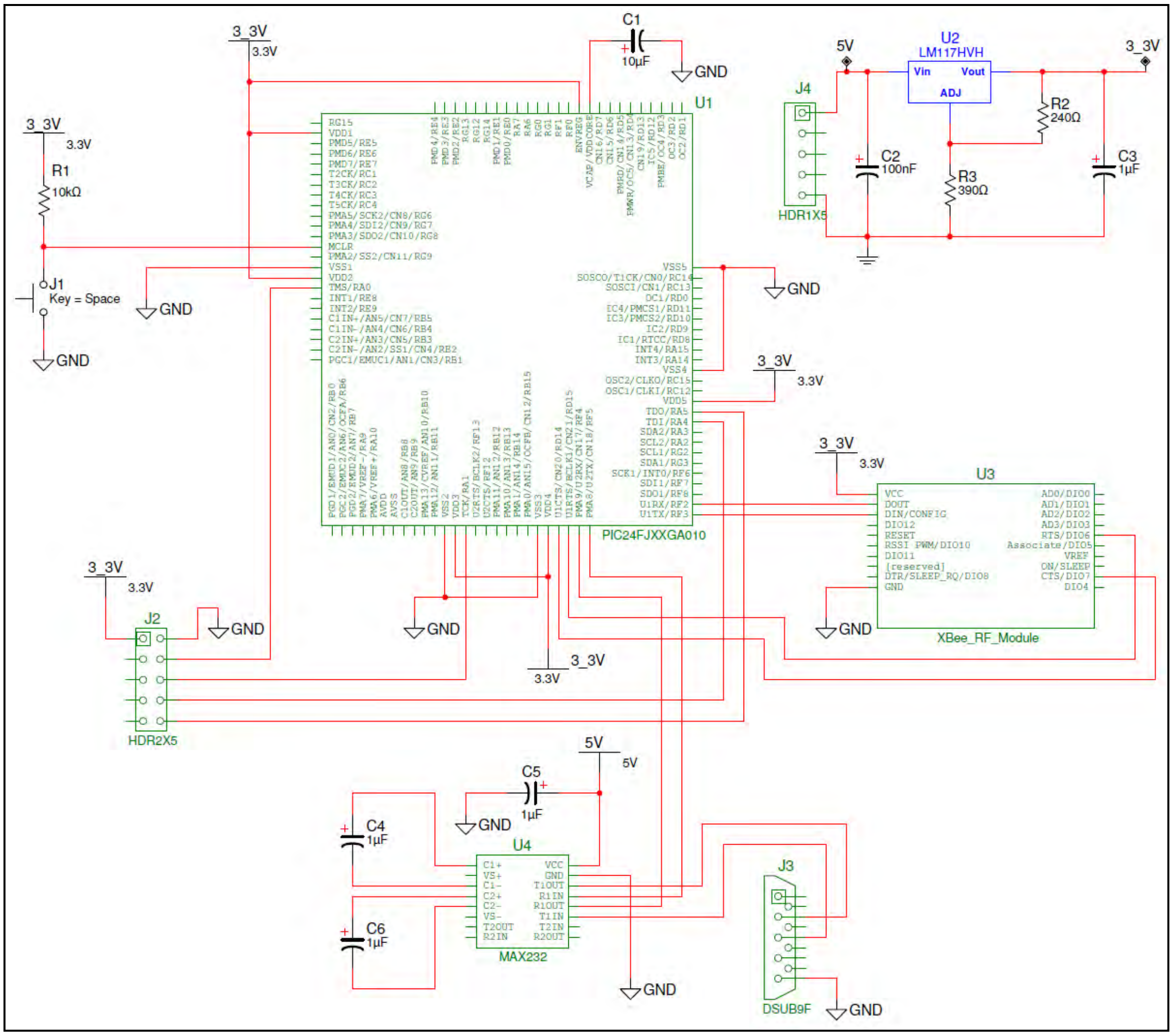


Figure 4: Original schematic for the Central Server Module

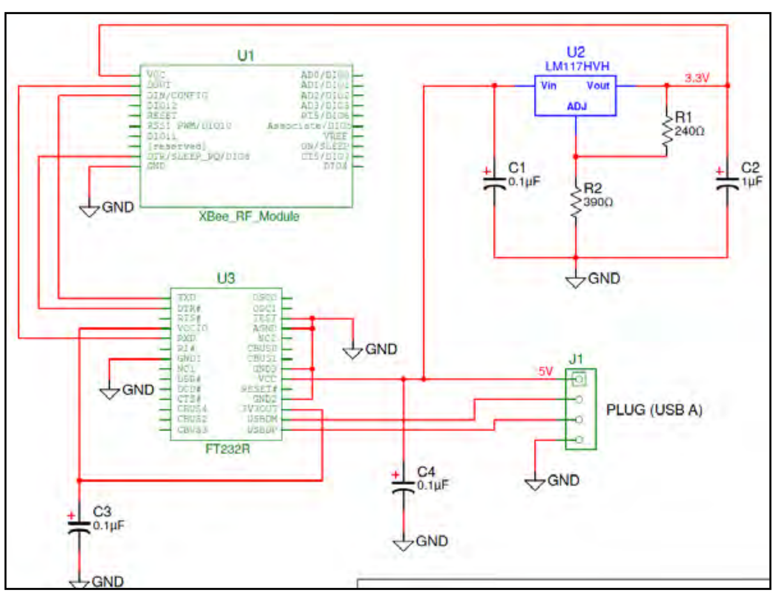


Figure 5: Revised schematic for the Central Server Module.

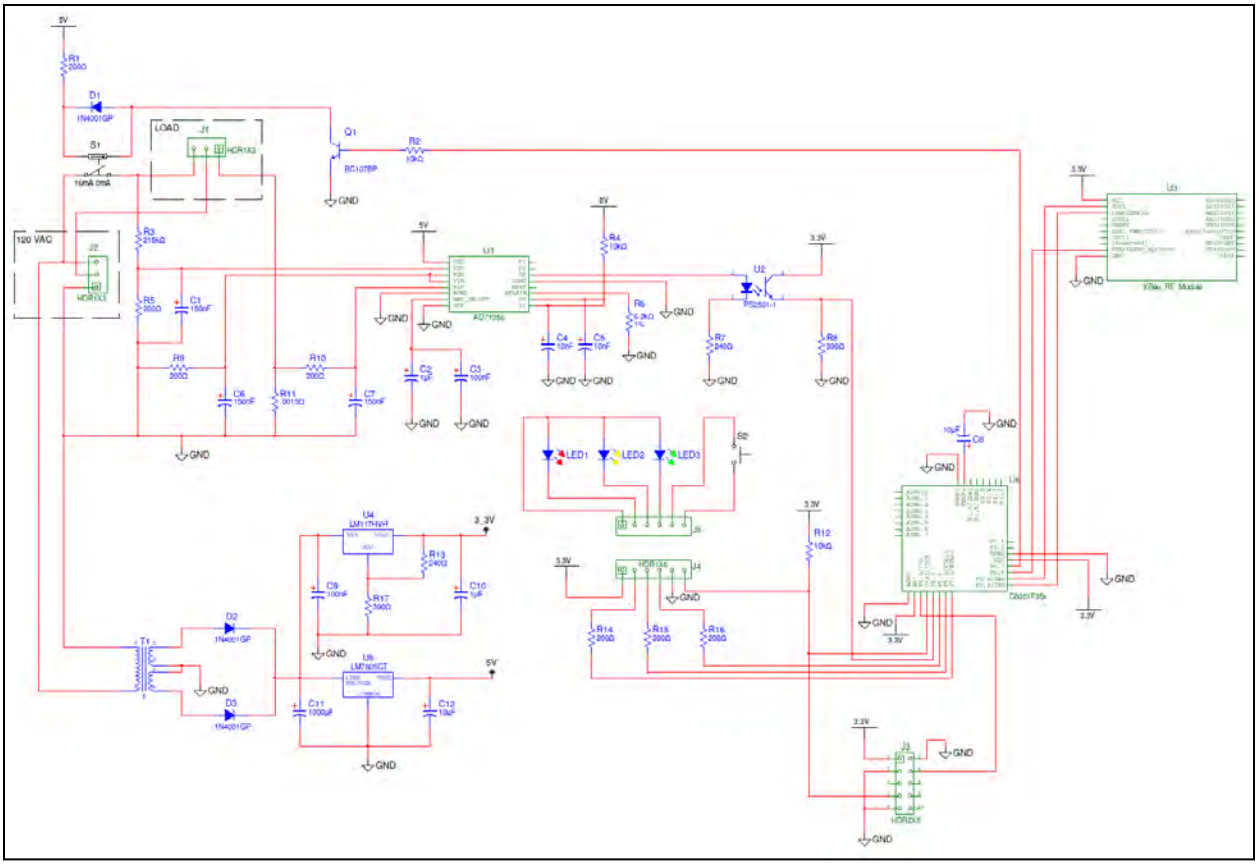


Figure 6: Revised schematic for the Measurement Node.

2.3 COMPONENT SELECTION AND PURCHASING

Once the schematics were completed, each of the actual components to be used in our design had to be selected. Several factors were considered when selecting components. The first consideration was selecting the component package to use for each part. Most of the electrical components in our design were selected in a surface mount package. This reduced the footprint required for each component and the overall space required on the printed circuit board. Several through-hole components were used when a comparable surface mount component was not available. Another consideration was the tolerance required for each component. Several resistors with a tolerance of ±1% were required in the circuit when accuracy was critical. Additionally, the power and voltage rating for each of the components was considered to make sure that none of components would become overstressed during use. Finally, cost and availability were considered. By using standard components, we were able to keep our cost low and we were also able to purchase components in the low quantities required for our project. The majority of our electrical components were purchased from Digikey [5] and node enclosure was purchased from Allied Electronics [2]. A complete list of parts and cost is discussed in section 4.

2.4 PRINTED CIRCUIT BOARD (PCB) LAYOUT FOR MEASUREMENT NODES

With the schematics completed and the components selected, the printed circuit board layout for the measurement node was created. The schematics for our design were created using National Instrument's Multisim software. All of the component packages were first updated in Mulitsim to convert the virtual components used for simulation into the actual components used for layout. The PCB layout was created using NI Ultiboard which uses the netlist created in Mulitsim to define all of the components and the connections between each component. After importing the components into Ultiboard, the components were placed on the board. Much consideration was used while placing components in order to reduce the length of traces without creating interferences with other components. Figures 7-9 below show the layout process in the different stages.

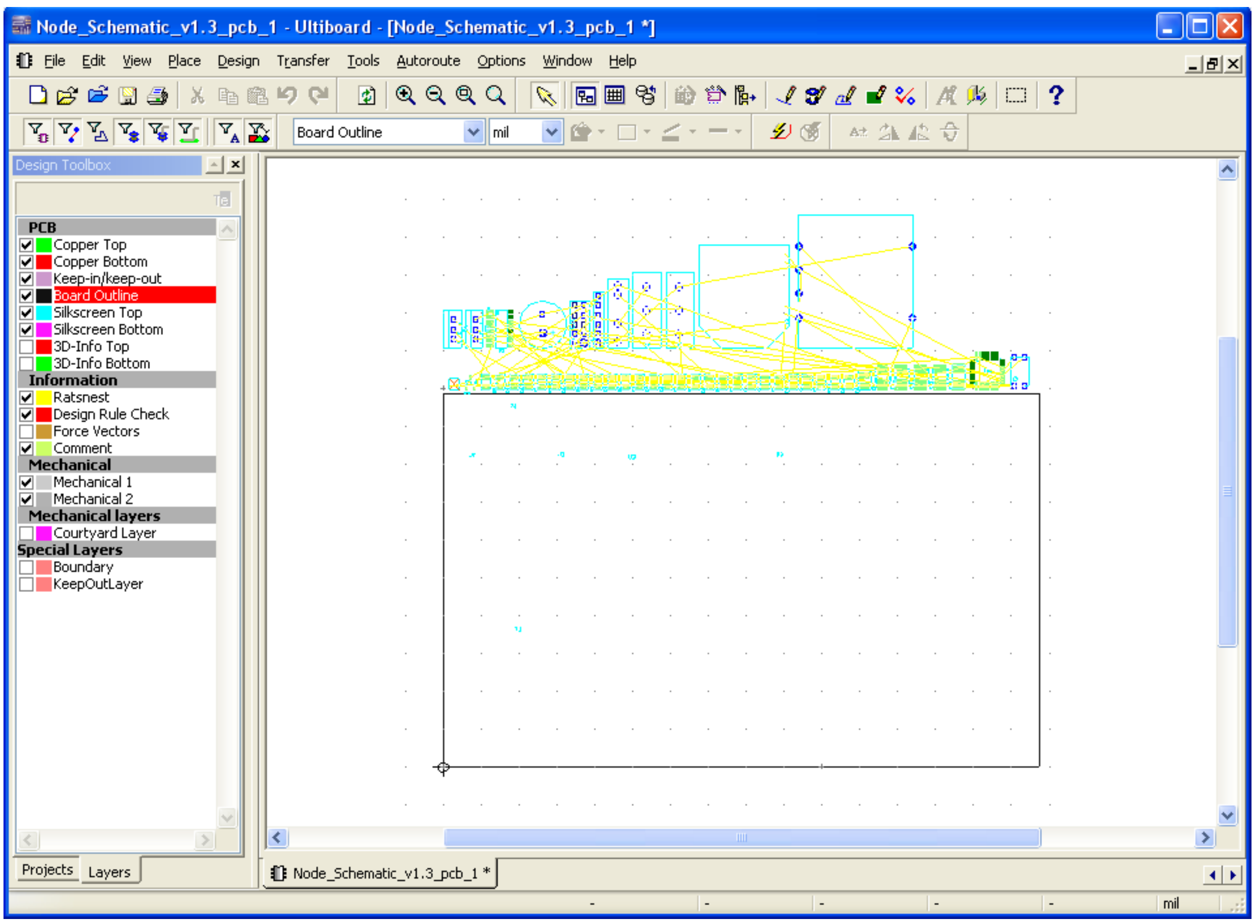


Figure 7: PCB layout for the Measurement Node before placement of components.

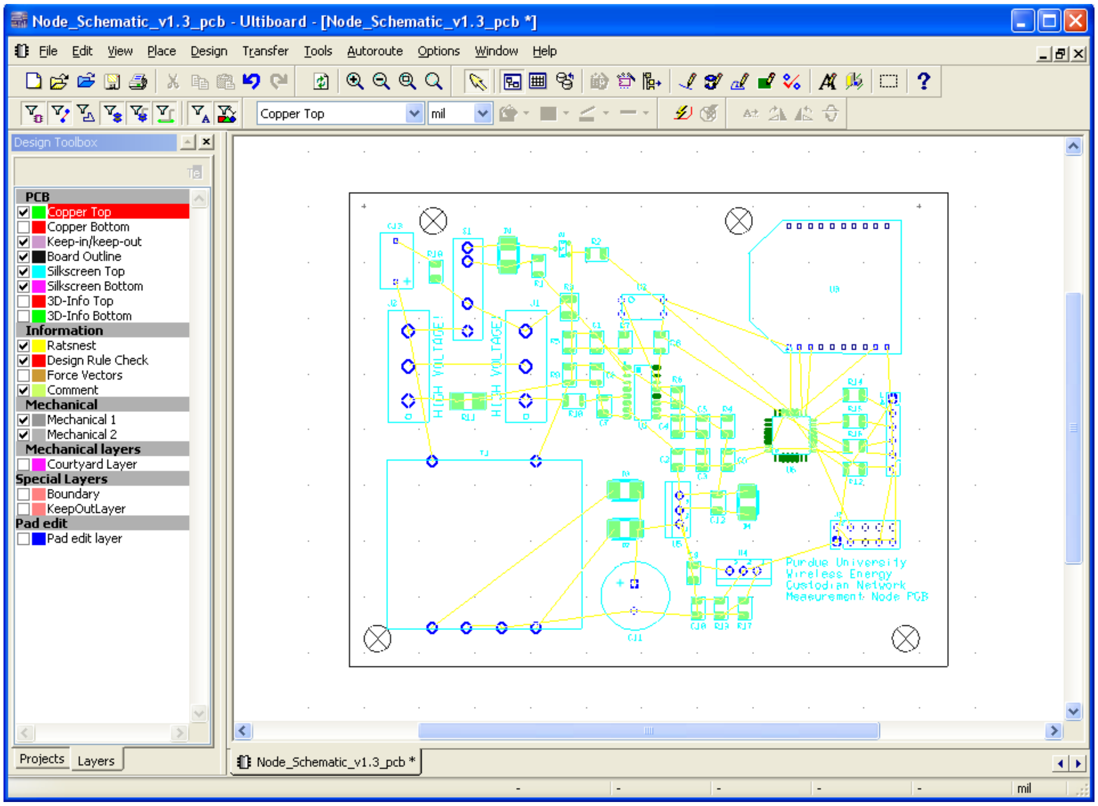


Figure 8: PCB layout for the Measurement Node after placement of components.

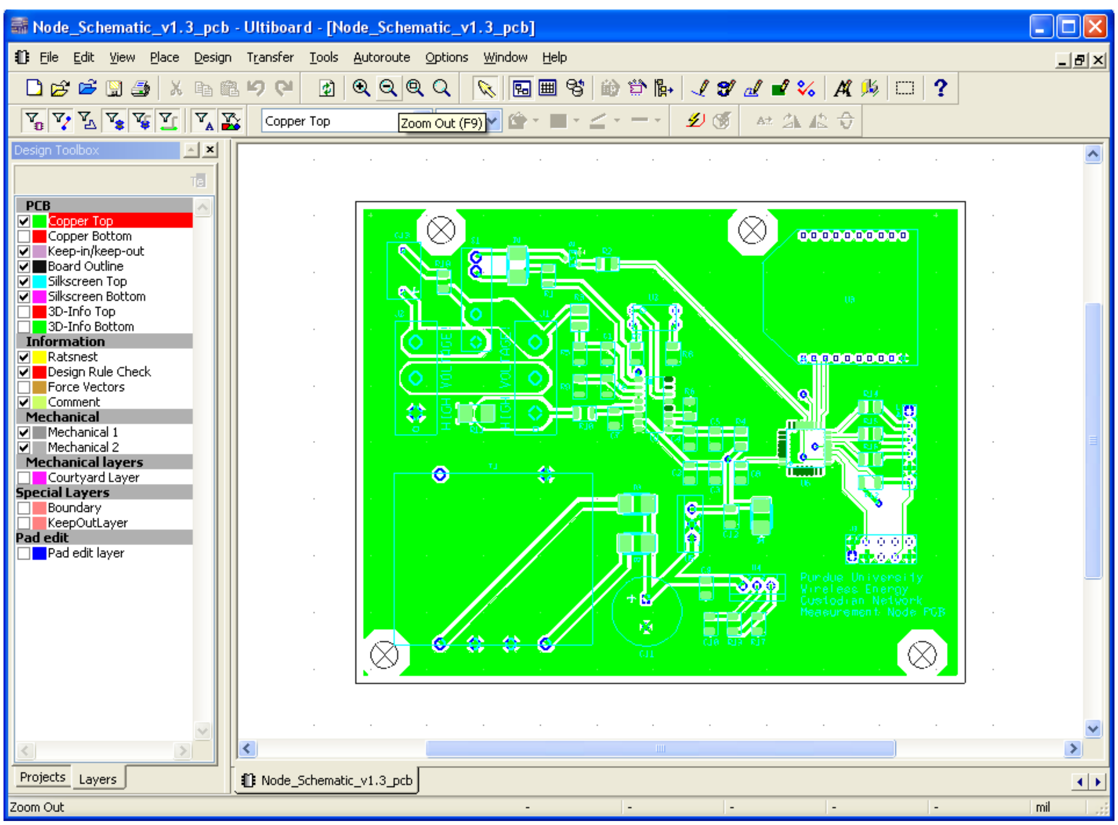


Figure 9: PCB layout for the Measurement Node after traces connected to each component.

2.5 ASSEMBLY OF MEASUREMENT NODES

After the PCB layouts were completed, the Gerber files were sent to Advance Circuits [7] to have the PCBs manufactured. The remaining components were purchased from Digikey and Allied Electronics. The openings for the node enclosure were machined using a Bridgeport vertical mill in the IPFW machining center. The printed circuit boards were assembled at the technology lab using the reflow oven for the surface mount parts and a soldering iron for the remaining through-hole components. The overall size of the node is 4.7” x 4.7” x 2.4” which is smaller than the original design constraint of 6” x 6” x 3”. The total cost for each measurement node was $112.79 (The detailed cost analysis of the measurement node can be found in Section 4.2). Figures 10-12 below show the assembly in progress.

Figure 10: Photograph of the Measurement Node enclosure after machining openings.

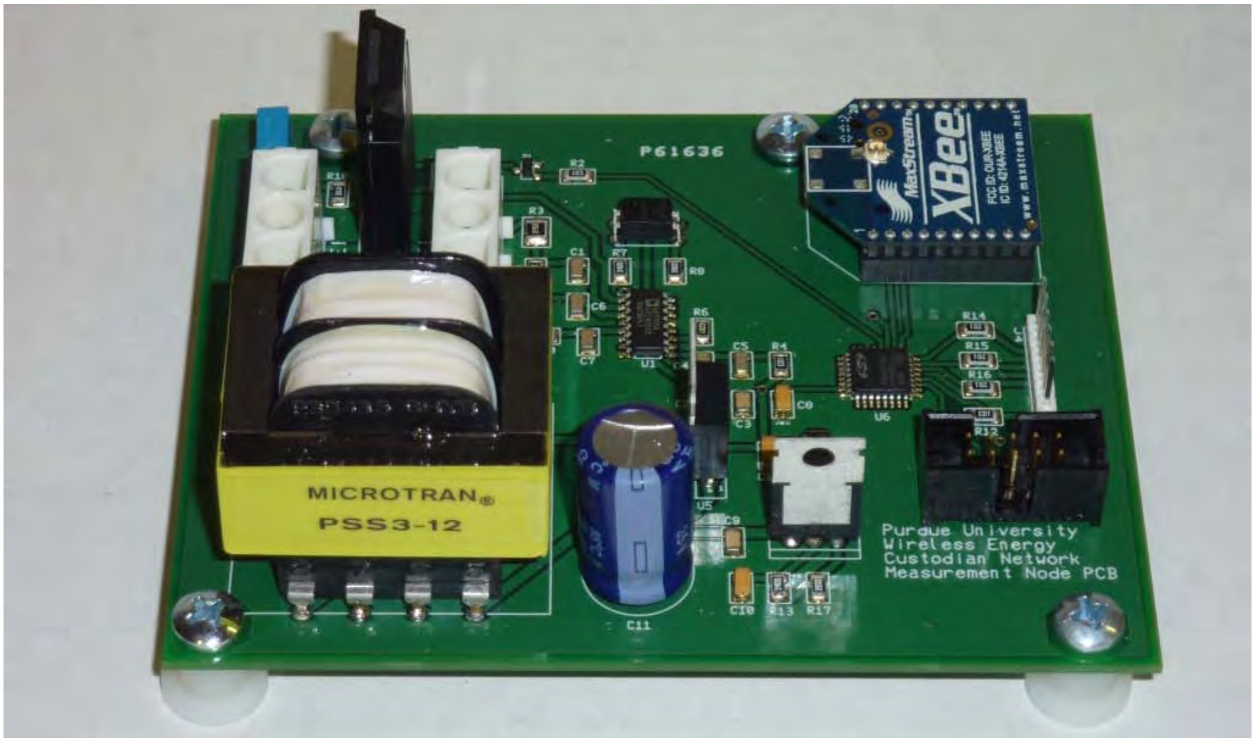


Figure 11: Photograph of the Measurement Node circuit board after assembly.

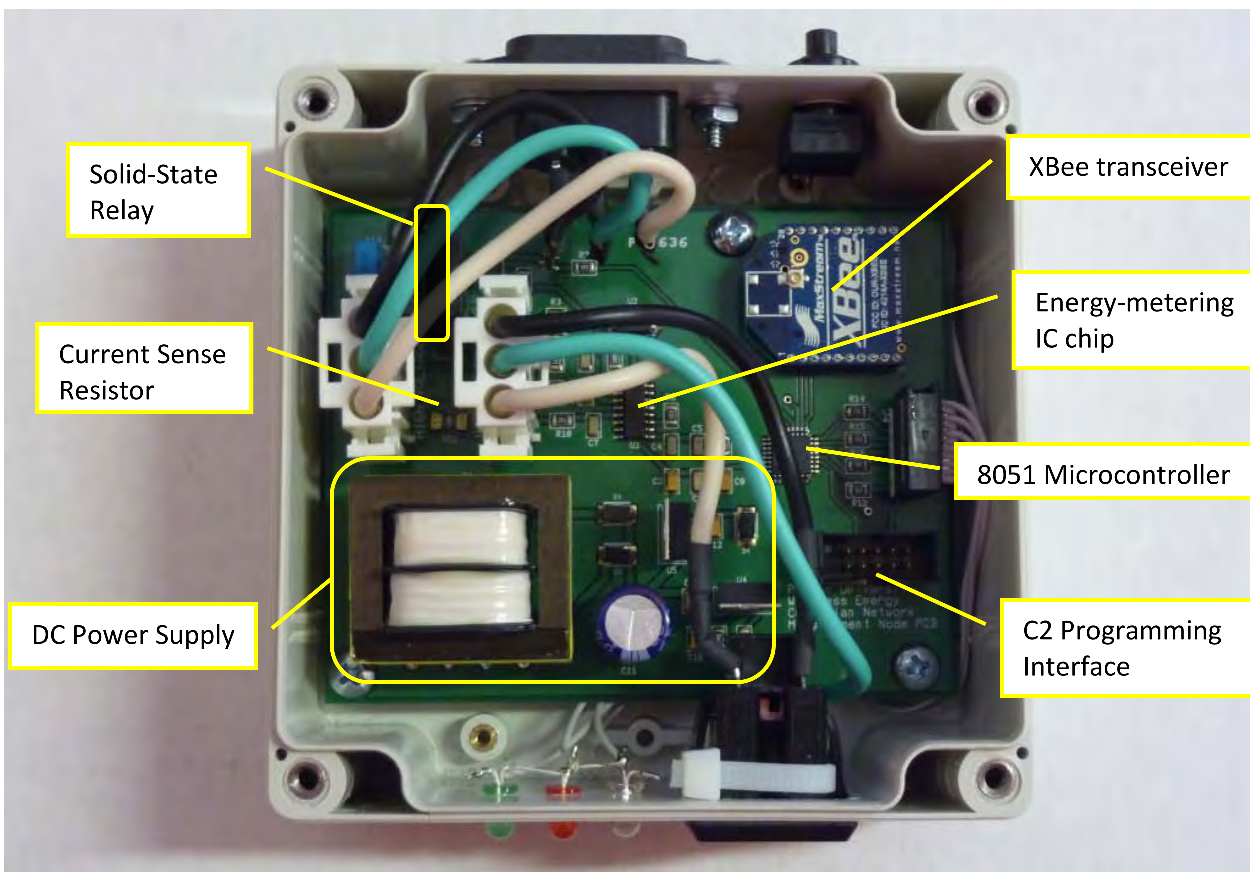


Figure 12: Photograph of the completed Measurement Node.

2.6 NODE SOFTWARE AND PROGRAMMING OF THE MEASUREMENT NODES

The node software was written and tested using the Keil uVision4 embedded development tool for the 8051 microcontroller. The ANSI C programming language was used to write our code, which was then compiled into a HEX file to program the microcontroller. The main function of the software is the collect measurements from the energy-metering IC chip, transmit the measurements to the CSM, receive control signals from the CSM, and control the power supplied to the connected appliance. The flowchart in Figure 13 illustrates the operation of the node measurement software. A complete copy of the measurement node software is included in the appendix.

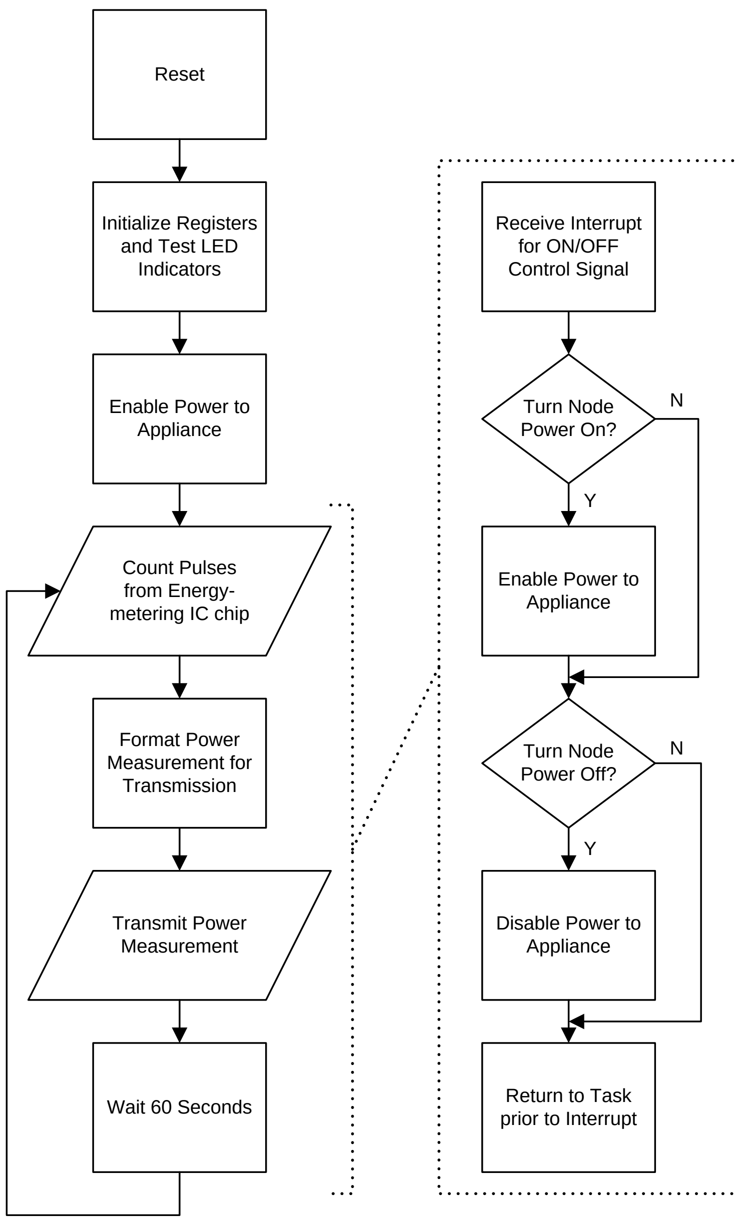


Figure 13: Flowchart for the measurement node software.

The node software was then loaded into the Silicon Labs C8051F350 microcontroller using the Silicon Labs USB Debug Adapter [9]. The adapter was connected to the C2 programming interface on the measurement node and the software was uploaded using Silicon Labs FLASH Programming Utility. Each measurement node has a distinctive version of the software that uniquely identifiers each node during transmissions with the CSM.

2.7 GRAPHICAL USER INTERFACE (GUI) SOFTWARE

The GUI software's main purposes are to display the power consumption data in real time and control the solid-state relays for each of the Nodes. To achieve these goals, the software is organized into two threads: the transmission thread and the user input thread.

1. The transmission thread will execute once per minute, as that is the rate at which the Nodes transmit their data. The transmission thread consists of the following steps (also shown in the flowchart in Figure 15):
    a. Open Serial Port -- This enables the software to use the USB Port on the computer and read in from the transceiver.
    b. Read Existing Data -- The software will wait until there is data available on the read buffer before continuing.
    c. Record Time Stamp -- The software records the time at which the data is read from the buffer to be recorded in the .CSV file and for energy consumption calculations.
    d. Parse Buffer Data –The data packet that the measurement nodes transmit to the CSM has the format shown in Figure 14 below. The data is read in as 2 bytes (or 16 bits). The lowest-order 5 bits of each byte represent the frequency count data received. The next bit is '1' to indicate the frequency count bits are the upper 5 bits of the 10-bit data or '0' to indicate the lower 5 bits. The following bit is used for Node identification. For our application, we refer to the Node as "Node 0" if this bit is '0', and "Node 1" if it is '1'. The last bit is unused, but could be used to expand the Node identification field and allow for 4 Nodes, or to increase the data resolution from 10 bits to 12 bits.
    e. Standby Power Consumption Control (SPCC) -- If the Standby Power Consumption Control is enabled, the system will either add the most recent measurement to the threshold calculation or increase the counter for the number of samples below the threshold. This is explained further in section 3 below.

Upper Byte (Node 0)									
Start	**D5**	**D6**	**D7**	**D8**	**D9**	**U/L**	**NID**	**-**	**Stop**
0						1	0	0	1
Lower Byte (Node 1)									
Start	**D5**	**D6**	**D7**	**D8**	**D9**	**U/L**	**NID**	**-**	**Stop**
0						0	1	0	1

Figure 14: Transmission Protocol

f. Append to .CSV File -- The most recent measurement is appended to the .CSV file in the format: “03/31/2011 19:06:28 PM, Node 0,19.16275,1000,”. The first field is the date and time of when the sample was received by the computer. The second field indicates the Node the data belongs to. The third field is the power value received. The last field indicates the time in milliseconds between the current sample and the sample before it.

g. Add Point to Display -- The power consumption data is added to the graph according to which Node it belongs to. The top graph shows data for Node 0, and the bottom graph shows data for Node 1.

h. Close Serial Port -- The serial port is then closed to save power.

i. Pause Thread -- The thread is allowed to “sleep” until the next value from the Nodes is expected.

2. The user input thread allows the software to asynchronously handle input from the user. Each time the user presses a button on the GUI, it produces a result.

   a. Menu -> Exit -- Selecting this button will pop up a message to have the user confirm their choice to exit. If the choice is confirmed, the program exits.

   b. CSM -> Edit CSM Port -- Selecting this button will pop up a message asking the user to give the port name for the CSM, after first closing the open port (if any). The software will then attempt to open the designate port.

   c. Monitor -> Start Monitor -- This button will verify that the serial port is valid, then start the transmission thread. If the serial port isn't valid, it will prompt the user to enter a valid serial port.

   d. Monitor -> Stop Monitor -- This button aborts the transmission thread and closes the serial port without clearing the graph or exiting the program.

   e. Node -> Enable/Disable Power -> Node 0/1 -- This button changes the control byte sent out by the CSM to the Nodes.

f. Node -> Enable/Disable Power Save Feature -> Node [0/1] -- This button enables or disables the power save feature for a specific node. See section 3 below.

g. Display -> Clear Data -- This button will cause the graphs to clear (including the axes). When the next data measurement is received, the graph will re-appear.

h. Display -> Calculate Total Energy -- This button aborts the transmission thread and opens a window prompting the user to select a date range. See section 4 below.

3. Power Save Feature -- The purpose behind the power-save feature is to reduce power consumption by devices that go into a “standby” mode. When this feature is first enabled, the software gathers the next 10 readings it receives for that Node and averages them. The average is multiplied by 110% and set as the threshold for that Node. Once the device has operated for 30 consecutive minutes below the threshold, power to the device is shut off via the control byte. A value above the threshold resets the count of “below threshold” values.

4. Calculate Total Energy -- When this window is opened, the user is asked to provide a “start date” and “end date” for the software to calculate the total energy consumed by both nodes. To calculate the energy, the software reads in historical data from the .CSV file and multiplies the time duration values by the power usage value. These are summed up and converted from Joules to kilowatt-hours and then displayed to the user.

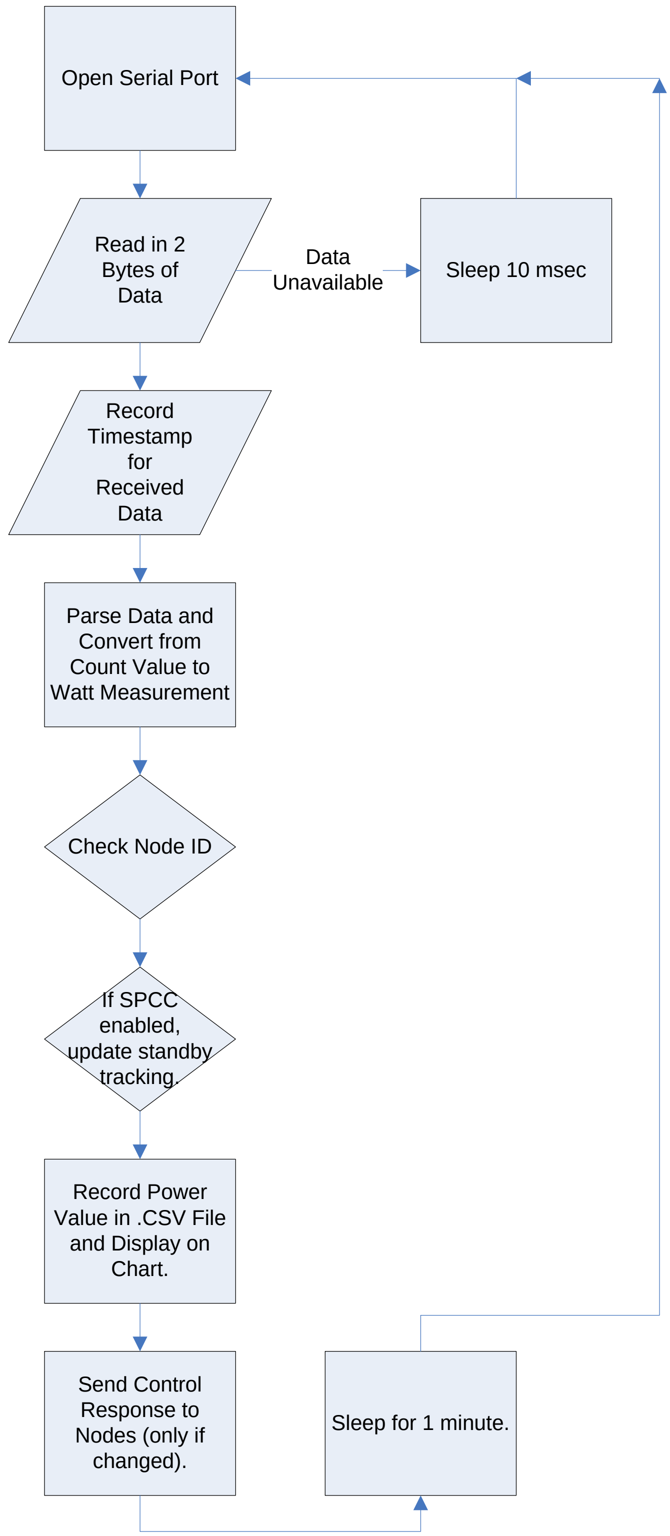


Figure 15: Flowchart for Transmission Thread

Figures 16 below shows the user defining the CSM port they desire to use. This is the first step to using the GUI software.

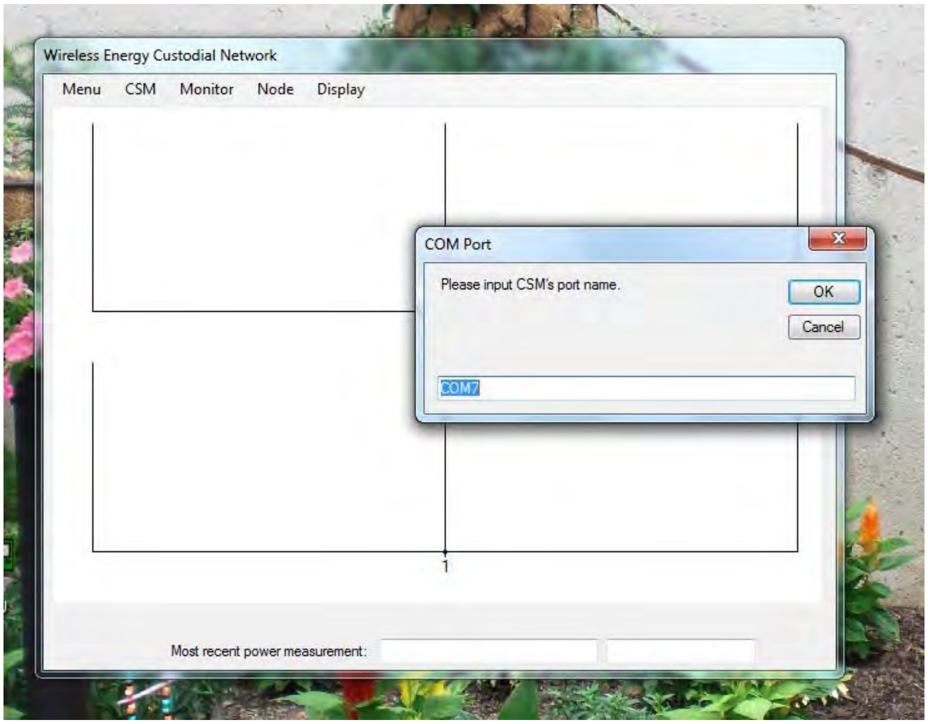


Figure 16: Editing CSM Port Name

Figure 17 below shows the user displaying the energy consumption for the time period on April 13th that the Energy Monitor was running. This energy consumption calculation can be performed for any time period.

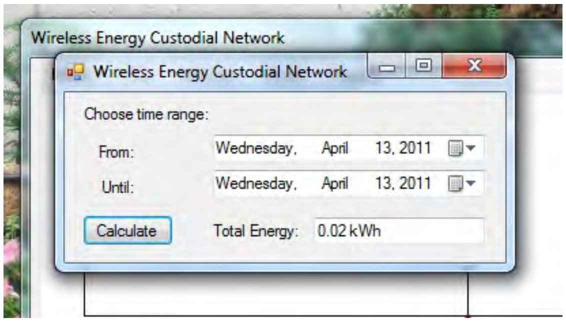


Figure 17: Calculating Energy Consumption over a time period

SECTION 3: TESTING AND RESULTS

OVERALL TESTING PROCEDURE

For our design we tested each of the major components required for proper operation of the nodes and the central server module. Once each of these components was tested, the nodes and the CSM were assembled and readied for a complete system test. This included measuring the power consumption at the node, wirelessly transmitting the information to the CSM, rendering the information on the GUI in real time, and transmitting a control signal back to the node if necessary. The components which were tested were the AC-DC power supply, the solid state relay, the optocoupler, the energy metering IC chip, the transceiver, and the GUI/CSM.

3.1 ANALOG DEVICES -- ENERGY METERING IC CHIP - AD71056ARZ

Testing Procedure:
To test the functionality of the AD71056 the frequency measurement at the output of the AD71056 was recorded at different levels and compared to the theoretical values given by equation 1 below [1].

Power consumption (in Watts):

$$P = \left(\frac{V_{rms} I_{rms}}{f_{\max}} \right) f_{CF} = \left(\frac{125 \times 15}{2866} \right) f_{CF} = \left(\frac{1875}{2866} \right) f_{CF} \qquad [1]$$

Table 2 below shows the results for several load values that were tested.

Table 2: Test values for the frequency output at CF pin for several power consumption values.

Voltage (rms)	Current (rms)	Power	Frequency (theoretical)	Frequency (actual)	Error
125 V	0	0 W	0 Hz	0 Hz	-
125 V	96 mA	12 W	18 Hz	25 Hz	+38 %
125 V	120 mA	15 W	23 Hz	34 Hz	+48 %
125 V	464 mA	58 W	87 Hz	135 Hz	+55 %
125 V	5.7 A	710 W	1085 Hz	1610 Hz	+48 %
125 V	6.0 A	750 W	1150 Hz	1640 Hz	+43 %
125 V	8.3 A	1040 W	1590 Hz	2400 Hz	+51 %

Testing Results:
These results indicate a scaling error for all of the measurements likely due to a larger-than-expected voltage across the current sensing inputs of the AD71056. The current in the circuit is measured with a low impedance 1.5 mΩ resistor. This, coupled with the additional 0.7 mΩ of resistance through the printed circuit board traces, would account for the error in the above measurements. To determine the correction factors needed to reduce the scaling error the test values were plotted as in Figure 18for node 0.

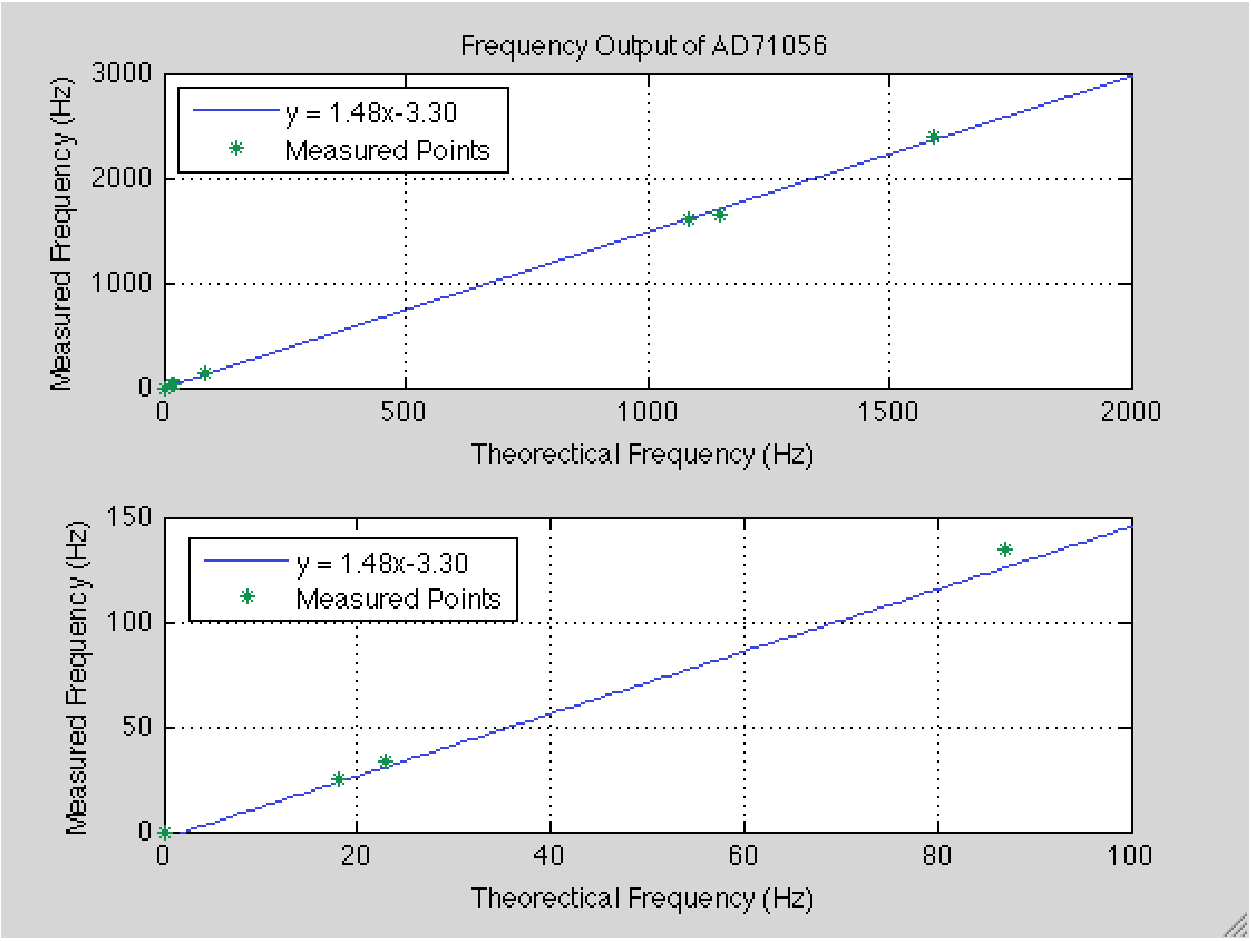


Figure 18: Test values for the frequency output at CF pin for several power consumption values. The correlation coefficient for this data is 0.9994, which suggests the relationship is linear. The linear regression of the data points results in the equation: [$y = 1.48x – 3.30$]. The slope (1.48) represents a scaling error for the entire range of power measurements. The y-intercept (-3.30) represents the zero offset error at lower power measurements.

This was done for both nodes and the result is shown in Figure19.

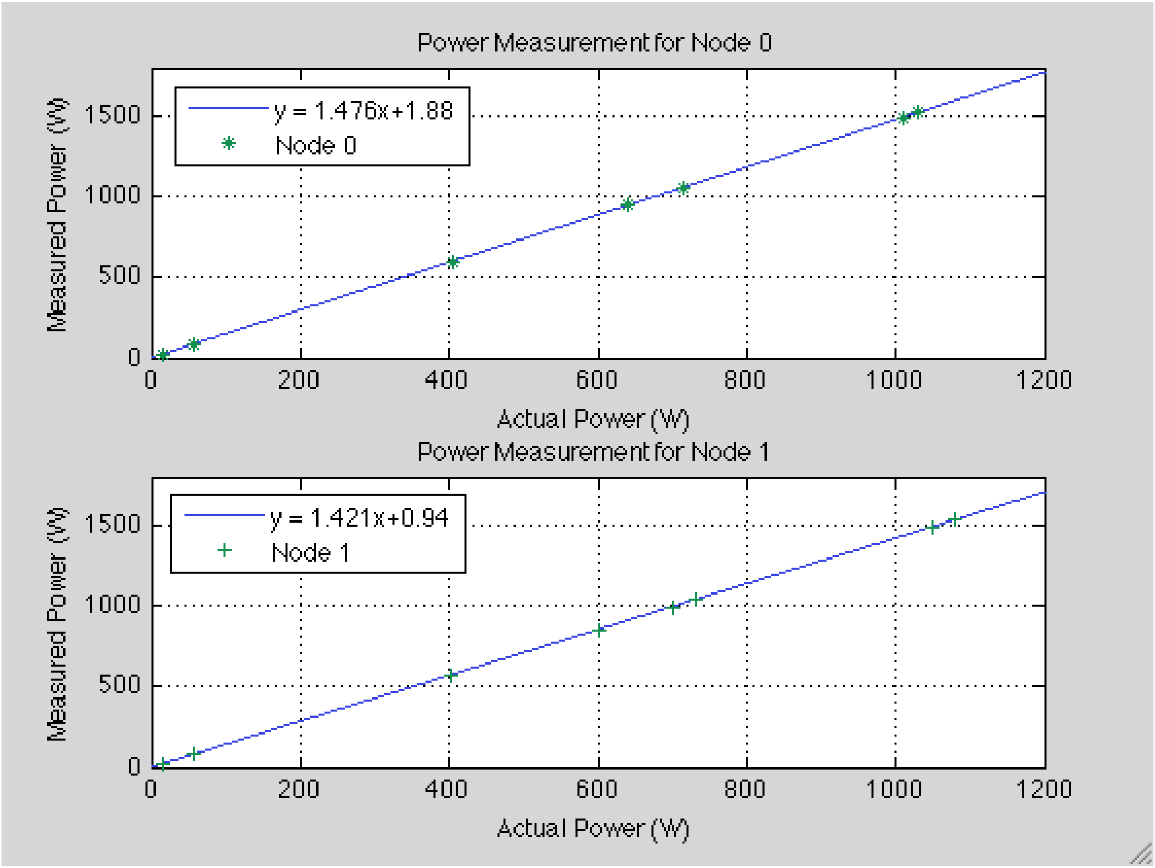


Figure 19: Test values for the frequency output at CF pin for several power consumption values for both nodes 0 and 1.

The slope of the line for each node is the correction factor. For node 0, the measured value is divided by 1.476 to get the corrected value. For node 1, the measured value is divided by 1.421. This error is caused the additional resistance in the copper trace to the energy-metering chip.

$$Node\ 0: (1.5m\Omega)(1.476) = 2.21m\Omega, an increase of\ 0.71m\Omega.$$
$$Node\ 1: (1.5m\Omega)(1.421) = 2.13m\Omega, an increase of\ 0.63m\Omega.$$

There may be other small contributing factors to this error but the majority is caused by this added resistance.

3.2 SHARP ELECTRONICS - SOLID-STATE RELAY - S216802F

The solid state relay is rated for an absolute maximum of 50mA forward current at the input pin. The input signal current required for an ON state, according to the data sheet for the relay, is between 16 and 24mA. The current required for an OFF state is between 0 and 0.1mA. Because of this, we tested currents ranging from 0 through 24mA to see what state the solid state relay would be operating in.

Testing Procedure:
The mode of operation for the solid state relay is determined by the amount of current through the input pin of the relay. We tested the relay by incrementally varying this current from 0 to 24mA to determine the specific amount of current needed to switch the solid state relay from the off mode to the on mode.

Testing Results:
The results from the testing are summarized below in Table 3.

Table 3: Test values for the solid-state relay.

Input Signal Current	**Relay State (theoretical)**	**Condition (actual)**
0 mA	Off	Off
0.1 mA	Off	Off
1 mA	Unknown	Off
1.9 – 2.0 mA	Unknown	Unstable state, changes between on-off
10 mA	Unknown	On
16 mA	On	On
20 mA	On	On
24 mA	On	On

From our testing, we determined that

- Input current of less than 1.9mA results in the device attached to the relay being turned off
- Input current between 1.9mA and 2.0mA results in unstable device operation (e.g. flickering light bulb)
- Input current greater than 2.0mA results in the device operating as normal.

3.3 NEC - OPTOCOUPLER - PS2501-1-L-A

Testing Procedure:
The output pin of the AD71056 was connected to the PS2501 optocoupler [4]. An oscilloscope measured the voltage levels at the outputs of both the AD71056 and the PS2501 to ensure the optocoupler was generating a waveform which was of the same frequency as the output of the AD71056 but with a lower voltage, as shown in Figures 20-23 below.

Testing Results:
The yellow traces in the resulting figures are the outputs from the AD71506 while the blue traces are the outputs from the PS2501.

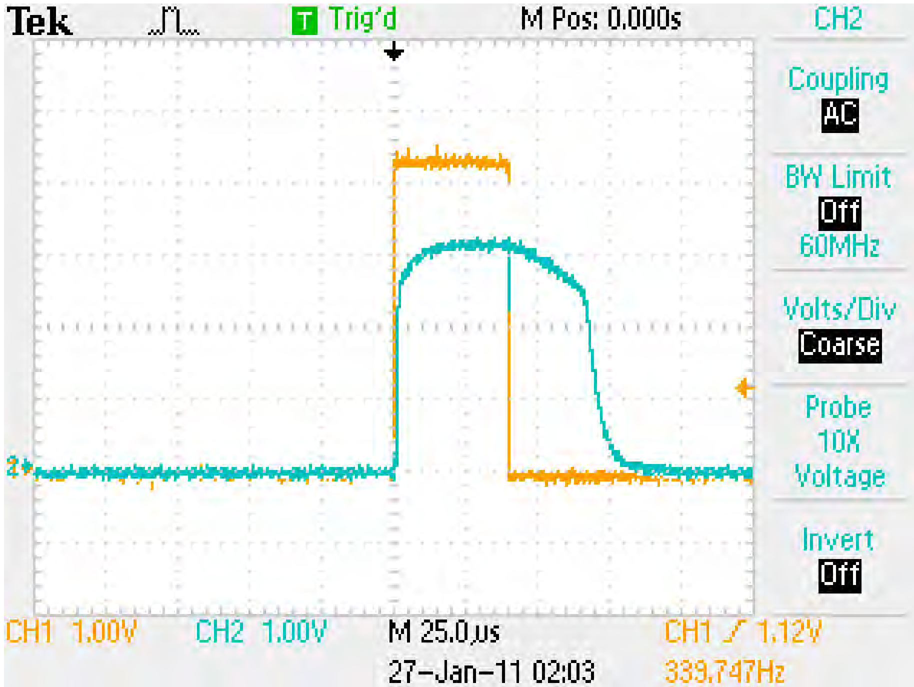


Figure 20: Yellow trace, AD71056 output, duration: 40 µs, amplitude: 4.8-4.9 V
Blue trace, optocoupler output, duration: 75 µs, amplitude: 3.1-3.2 V

Figure 21: AD71056 output waveform to be sent to C8051F342 microcontroller, frequency: 25 Hz

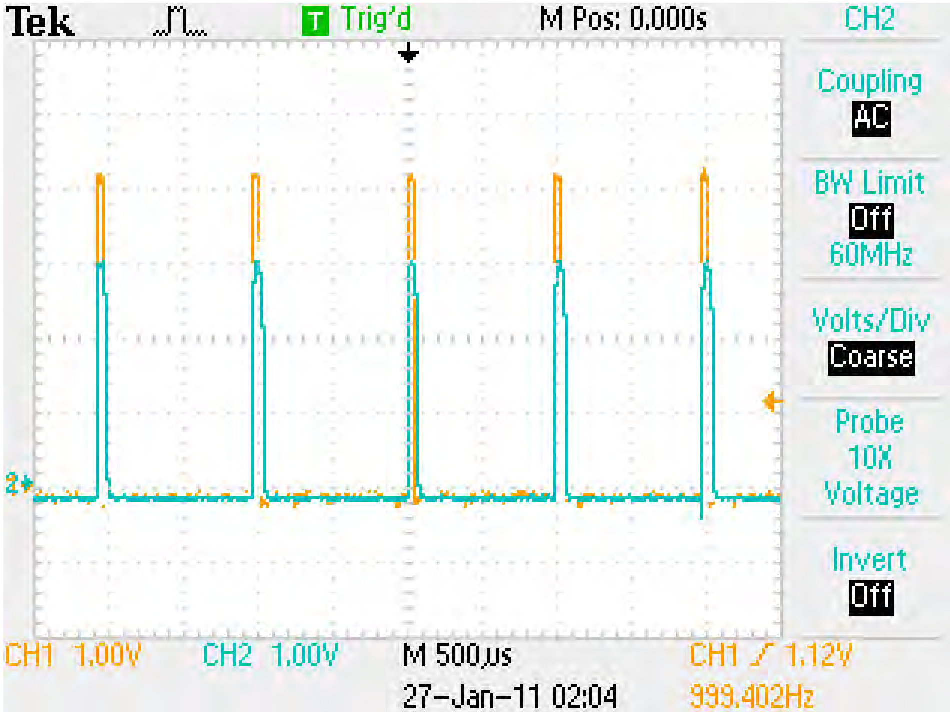


Figure 22: AD71056 output waveform to be sent to C8051F342 microcontroller, frequency: 999 Hz

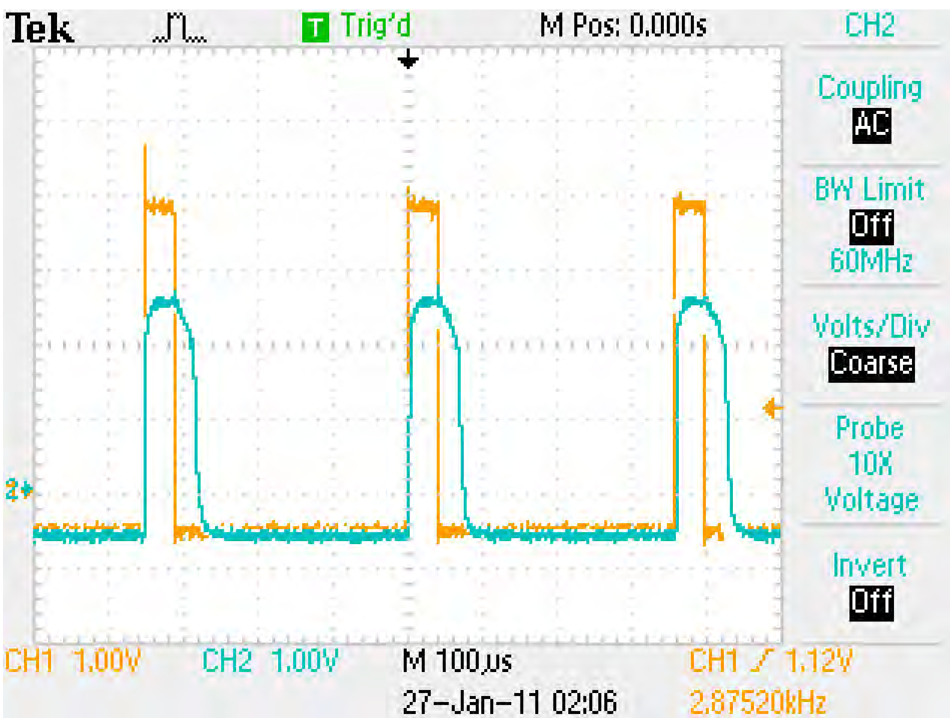


Figure 23: AD71056 output waveform to be sent to C8051F342 microcontroller, frequency: 2875 Hz

The PS2501 optocoupler works well for all of the operational frequencies used in this project, 1Hz- 3kHz.

3.4 FAIRCHILD SEMICONDUCTOR - POWER SUPPLY - LM7805CT-ND/LM317T

Testing Procedure:
The testing of the AD-DC power supply includes a 5-volt source and a 3.3-volt source. The voltage outputs from the power supply were measured while the current load connected to the outputs was varied. This variation was generated using a decade resistor box to adjust the load resistance for each voltage.

Testing Results:
The voltage outputs were measured as the load was adjusted and the results were compiled into Table4.

Table 4: Test values for the node power supply.

Load on 3.3 V source	Load on 5.0 V source	Voltage at 3.3 V source	Voltage at 5.0V source
0 mA (no load)	0 mA (no load)	3.37 V	4.97 V
20 mA (165 Ω)	0 mA (no load)	3.38 V	4.97 V
40 mA (83 Ω)	0 mA (no load)	3.38 V	4.97 V
60 mA (55 Ω)	0 mA (no load)	3.38 V	4.97 V
80 mA (41 Ω)	0 mA (no load)	3.38 V	4.97 V
100 mA (33 Ω)	0 mA (no load)	3.38 V	4.97 V
0 mA (no load)	20 mA (250 Ω)	3.38 V	4.97 V
0 mA (no load)	40 mA (125 Ω)	3.38 V	4.97 V
0 mA (no load)	60 mA (83 Ω)	3.38 V	4.97 V
0 mA (no load)	80 mA (63 Ω)	3.38 V	4.96 V
0 mA (no load)	100 mA (50 Ω)	3.38 V	4.96 V
66 mA (50 Ω)	60 mA (55 Ω)	3.39 V	4.97 V
100 mA (33 Ω)	100 mA (50 Ω)	3.39 V	4.96V

The power supply functioned as expected, generating results which were close to the required 3.3V and 5V values. The average errors for the 3.3V and 5V power supply are 0.080769V and 0.03231V, respectively, which are within the tolerance of 2.5% and 0.65%.

3.5 WIRELESS SIGNAL TESTING

There are two main parts of testing the wireless communication: range testing and impediment testing. Range testing involves seeing how far in any direction the transceivers can communicate. Impediment testing involves testing the signal strength through large appliances, metal objects, brick walls, and other wireless signals.

Two types of testing were conducted, “range” testing and “impediment” testing. Range testing was testing how far the node and CSM could be placed relative to each other while still receiving a signal which was free of transmission errors. Impediment testing was testing in which the node or the CSM was placed in or around objects which could cause distortion or loss of signal (for example, metallic objects like stoves, refrigerators, etc.) The design requirement was for a range of 15 meters to allow for use inside of a standard home.

Testing Procedure:
The nodes and the CSM were taken to a large three story home with a walk-out basement in Fort Wayne, Indiana.

Testing Results:
The results of the range testing were as follows in Tables 5-7. For the purposes of this test the result of "Strong" is defined as no noticeable transmission problems while "Weak" is defined transmission reception was sporadic or not at all.

Table5: Lateral range testing

Single Floor		
Range (feet)	Impediments	Signal
5	None	Strong
10	None, Interior Brick Wall	Strong
15	None, Interior Brick Wall	Strong
20	None, Interior Brick Wall	Strong
25	None, Interior Brick Wall, Interior Wall	Strong
30	None, Interior Brick Wall, Interior Wall	Strong
35	None, Interior Brick Wall, Interior Wall	Strong
40	None, Interior Walls	Strong
45	None, Interior Walls	Weak

Table6: Vertical range testing

Multi Floor		
Node Floor	CSM Floor	Signal
Upstairs	Ground	Strong
Basement	Ground	Strong
Upstairs	Basement	Strong

Table7: Impediment testing

Impediments			
Node	CSM	Comment	Signal
Basement	Ground	Node Inside Washing Machine	Strong
Ground	Ground	Node Inside Oven	Strong
Ground	Ground	Refrigerator Between CSM and Node	Strong

Our testing concluded that the range requirement was satisfied and that no noticeable impairment to the signal occurred when transmitting near or in large metallic objects.

3.6 TESTING CSM/GUI

In order to test the GUI software, a space heater with a power rating of 1500 W was plugged into one of the Nodes and an LCD computer monitor rated at 20 W (without the computer) into the other Node. Each of the features and every button provided in the GUI was tested, in a variety of orders as an attempt to "break" the code. Invalid inputs and deadlock errors were solved using error trapping within the code. Figures 24-25 show the GUI screenshot of these tests.

Figure 24: Screenshot of GUI while Testing Heater (Top Graph) and Computer Monitor (Bottom Graph)

In order to test the standby power-save feature, we had to manually force an appliance to go into a pseudo-standby state. In order to do this, we plugged in the portable space heater to the Node and turned on the heat during the "threshold calculation" phase. After the threshold was calculated, we turned the heat off and just had the fan running on the heater. This resembles a drastic reduction in power consumption of an appliance. We were able to observe that after the desired number of samples below the threshold were received, power was shut off to the space heater.Figure 25 below shows the GUI after the auto-shutoff feature was enabled for the space heater.

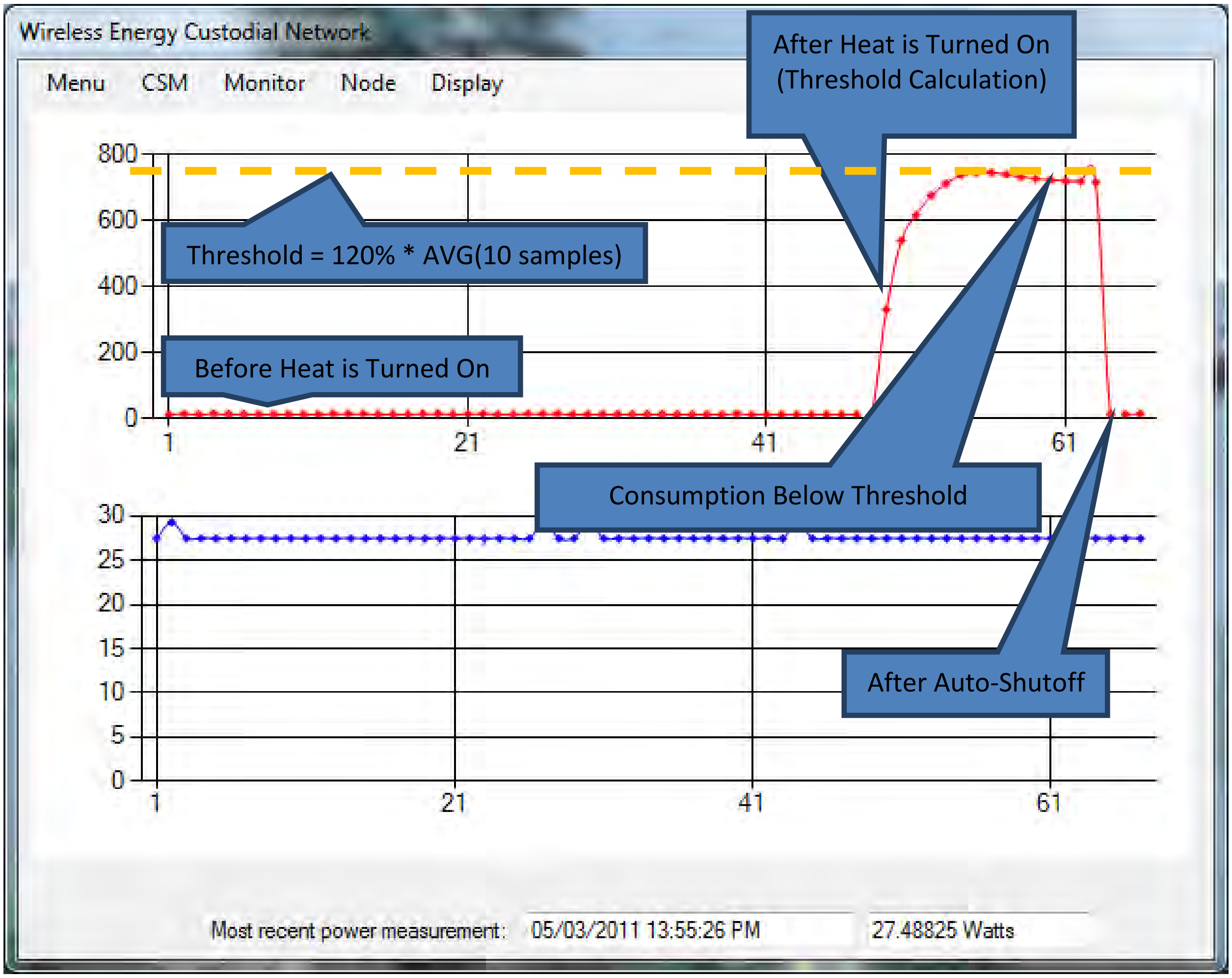


Figure 25: Screenshot of GUI after Heater (Top Graph) has Reached its Auto-Shutoff Point

As indicated by Figure 25, the heater was turned off automatically after the large threshold was calculated.

3.7 TESTING NODE SOFTWARE

The software for the node was tested through multiple tests to ensure that all the functionalities were successfully implemented.

Test 1:
One of the first tests was to see if the I/O ports could read the signal. First a function generator was used, and then the input signal from the energy metering IC was used. Both the function generator and the input signal from the energy metering IC were also used to see if the frequency counter implemented in the node software worked correctly

Results:
Reading the signal from the I/O ports and from the input signal from the energy metering IC was proven successful. The frequency counter was proven successful as well.

Test 2:
The microcontroller was connected to the Xbee transceiver so that the wireless communication between the node and the Xbee Explorer Dongle, which was connected to a computer, was tested.

Results:
The wireless communication between the node and the Xbee Explorer Dongle was proven successful.

Test 3:
The reset button was tested by pushing the reset button.

Result:
The reset button functionality was proven successful.

Test 4:
The output signals for the indicator LEDs were tested. The 'yellow' LED should turn on every time data was transmitted. The 'red' LED should light up once the attached device was turned off. The 'green' LED should light up when the device was on.

Results:
The correct on/off functionality of LEDs was proven successful.

Test 5:
The output signal for the solid-state relay was tested. High output when the connected device is on.

Results:
The solid-state relay functionality was proved to be working correctly. Whenever the CSM transmitted to the node to turn the device on/off, the device was able to turn on/off.

SECTION 4: COST ANALYSIS

4.1 CENTRAL SERVER MODULE

The cost to produce the CSM was calculated using two pricing structures. The first structure includes the total prototype cost to build a single module. This provided a good estimate of the cost when purchasing components in small quantities, such as for this project. The second pricing structure includes the total cost of each module when purchasing components in larger quantities. This will provide a good estimate of the unit cost when planning a low volume production run, such as 1000 units. The total cost estimate to build a single module is $69.50. This is reduced to $33.06 when purchasing components in bulk. Table 8 list the components required to build the CSM and the cost for each component. In place of building the CSM in our design, we were able to use aXbee transceiver connected to an Xbee Explorer Dongle from Sparkfun Electronics. The total cost for both items was $43.95.

Table 8: Cost for the Central Server Module.

ITEM DESCRIPTION	QUANTITY	REF DES	ITEM PRICE (SINGLE)	ITEM PRICE (BULK)	CSM COST (SINGLE)	CSM COST (BULK)
IC REG ADJ 1.5A	1	U2	$0.65	$0.22	$0.65	$0.22
CAPACITOR, TANTALUM	1	C2	$0.40	$0.14	$0.40	$0.14
CAPACITOR, CERAMIC	3	C1,C3,C4	$0.17	$0.04	$0.51	$0.12
RESISTOR - 1/4 W	1	R1	$0.10	$0.01	$0.10	$0.01
RESISTOR - 1/4 W	1	R2	$0.10	$0.01	$0.10	$0.01
XBEE RF MODULE W/CHIP ANT	1	U1	$19.00	$19.00	$19.00	$19.00
IC USB FS SERIAL UART	1	U3	$4.50	$2.85	$4.50	$2.85
CONN PLUG USB A 4POS SMD R/A	1	J1	$1.24	$0.71	$1.24	$0.71
ENCLOSURE	1		$10.00	$5.00	$10.00	$5.00
PRINTED CIRCUIT BOARD	1		$33.00	$5.00	$33.00	$5.00
				TOTAL UNIT COST	$69.50	$33.06

4.2 MEASUREMENT NODES

The cost to produce the each measurement node was also calculated using the same two pricing structures described above. The total cost estimate to build a single node is $112.79. This is reduced to $59.12 when purchasing components in bulk. Table 9 list the components required to build each node and the cost for each component. A complete bill of materials for the CSM and the each node is included in the appendix.

Table 9: Cost for each measurement node.

ITEM DESCRIPTION	QUANTITY	REF DES	ITEM PRICE (SINGLE)	ITEM PRICE (BULK)	NODE COST (SINGLE)	NODE COST (BULK)
IC ENERGY METERING 1 PHASE	1	U1	$2.63	$1.47	$2.63	$1.47
OPTOCOUPLER 1CH TRANS	1	U2	$0.52	$0.15	$0.52	$0.15
RESISTOR - 1 W	1	R11	$1.18	$0.31	$1.18	$0.31
IC REGULATOR - 1 A, 5 V	1	U5	$0.68	$0.20	$0.68	$0.20
IC REG ADJ 1.5A	1	U4	$0.65	$0.22	$0.65	$0.22
POWER TRANSFORMER	1	T1	$4.23	$2.03	$4.23	$2.03
DIODE GEN PURPOSE 50V 1A	4	D1,D2,D3,D4	$0.36	$0.05	$1.44	$0.20
CAPACITOR, ALUMINUM	1	C11	$0.82	$0.23	$0.82	$0.23
CAPACITOR, TANTALUM	2	C8,C12	$0.38	$0.13	$0.76	$0.26
CAPACITOR, TANTALUM	2	C2,C10	$0.40	$0.14	$0.80	$0.28
CAPACITOR, CERAMIC	2	C4,C5	$0.10	$0.02	$0.20	$0.04
CAPACITOR, CERAMIC	2	C3,C9	$0.17	$0.04	$0.34	$0.08
CAPACITOR, CERAMIC	3	C1,C6,C7	$0.36	$0.04	$1.08	$0.12
CAPACITOR	1	C13	$0.48	$0.25	$0.48	$0.25
RESISTOR - 1/4 W	2	R7,R13	$0.10	$0.01	$0.20	$0.02
RESISTOR - 1/4 W	2	R8,R17	$0.10	$0.01	$0.20	$0.02
RESISTOR - 1/4 W	1	R6	$0.10	$0.01	$0.10	$0.01
RESISTOR - 1/4 W	7	R1,R5,R9, R10,R14,R15,R16	$0.06	$0.01	$0.42	$0.07
RESISTOR - 1/2 W	1	R3	$0.52	$0.04	$0.52	$0.04
RESISTOR - 1/4 W	2	R2,R4,R12	$0.06	$0.01	$0.12	$0.02
RESISTOR - 1/4 W	1	R18	$0.06	$0.01	$0.06	$0.01
CONN HEADER 3POS VERT .250 TIN	2	J1,J2	$0.59	$0.35	$1.18	$0.70
MICROCONTROLLER	1	U6	$7.20	$5.78	$7.20	$5.78
XBEE RF MODULE W/CHIP ANT	1	U3	$19.00	$19.00	$19.00	$19.00
LED (RED)	1	LED 1	$0.12	$0.06	$0.12	$0.06
LED (YELLOW)	1	LED 2	$0.23	$0.11	$0.23	$0.11
LED (GREEN)	1	LED 3	$0.17	$0.08	$0.17	$0.08
RELAY SSR 240VAC 16A	1	S1	$5.30	$3.10	$5.30	$3.10
PUSH BUTTON SPST N/O MOM (RESET)	1	S2	$1.45	$0.54	$1.45	$0.54
JTAG HEADER (2x5 - 0.1" SPACING)	1	J3	$0.94	$0.38	$0.94	$0.38
LED/RESET HEADER (1x6 - 0.1 SPACING)	1	J4	$0.29	$0.14	$0.29	$0.14
LED/RESET WIRING CONNECTOR	1	J5	$0.33	$0.33	$0.33	$0.33
TRANSISTOR NPN 45V	1	Q1	$0.38	$0.04	$0.38	$0.04
OUTLET PWR NEMA 5-15R	1		$1.04	$0.34	$1.04	$0.34
IEC 320-C14 AC RECEPTACLE	1		$1.13	$0.37	$1.13	$0.37
CORD 16AWG 3COND M/F BLK 79" SJT	1		$6.10	$2.63	$6.10	$2.63
CONN PLUG 3POS 94V-0 UNI-MATE	2		$0.40	$0.19	$0.80	$0.38
CONN PIN 20-14AWG TIN UMNL	6		$0.12	$0.02	$0.72	$0.12
ENCLOSURE; ABS PLASTIC; 4.7 x 4.7 x 2.4in	1		$13.98	$12.39	$13.98	$12.39
PRINTED CIRCUIT BOARD	1		$33.00	$5.00	$33.00	$5.00
XBEE SOCKET (1x10 - 2 mm SPACING)	2		$1.00	$0.80	$2.00	$1.60
				TOTAL UNIT COST	$112.79	$59.12

4.3 DEVELOPMENT COSTS

The additional development costs were kept to a minimum by using a Silicon Labs development board from a previous project. Our group was able to use this development board to design the software and test the 8051 microcontroller used in each node. Several items were purchased to build a working measurement node on a breadboard. This allowed our group to test each of the subsystems before designing and building the final prototype. This included purchasing a residential receptacle box with a cord as a point to connect an appliance and several leaded electrical components needed for the breadboard circuit. The additional total cost for these items was approximately $25.

SECTION 5: EVALUATION AND RECOMMENDATION

5.1 IMPROVEMENTS AND RECOMMENDATIONS

- Solid State Relay:
  Discovered during testing, running an appliance that consumes large amounts of power can make the relay overheat, which may lead to product failure if the product is left on for too long. A solution to this would be to add a heat sink to the relay. The relay has space for a heat sink to be added which would allow the relay to cool down during extended periods of high power consumption.
- Reduce incoming current signal to the AD71056:
  This can be achieved by using a copper trace with 1.5mΩ of resistance in place of the1.5mΩ current sensing resistor coupled with the additional 0.7 mΩ of resistance through the printed circuit board trace.
- History Graph Integration into the GUI Software:
  This functionality can be added to future versions of the GUI software to enhance the number of options for the user. We were unable to implement this functionality during our time allowance because the way we tried to do it didn't work. Using C#, we attempted to keep separate sets of data points based on whether the data was from the past hour, day, or week. Then, based on the users' selection, the requested set of data was displayed. We ran into issues with attempting to dynamically change the set of data points, forcing us to remove this functionality.
- Power Reduction for Node:
  The node currently consumes 1.5W while transmitting information to the CSM, and 1W while idle. Possible sleep states could be looked into for the transceiver in the node to allow for less than 1W of power consumption while the node is not transmitting.
- Better Quality Transceivers:
  The maximum range of transmission between the nodes and the CSM is around 40 feet. The transceiver could be upgraded in order to increase this distance or the conscientious user can place the CSM near the center of their home in order to maximize the current transceiver capabilities.
- Data Resolution and Node Expandability:
  Our current design sends two bytes of data between the nodes and the CSM. This transmission protocol allows for each byte to contain the following: 2 bits of node identification (4 nodes possible), 1 bit of byte identifier (2 bytes possible), and 5 bits of data (10 bits of data resolution). Adding more bytes per transmission can allow for better data resolution or a larger number of nodes in the network. Four bytes could increase the data resolution to 16 bits. The bytes could be set up as follows: 2 bits of node identification (4 nodes possible), 2 bit of byte identifier (4 bytes possible), and 4 bits of data (16 bits of data resolution).
- Database Integration into the GUI Software:
  By integrating a database system into the GUI software faster data usage can be obtained. This usage includes data retrieval, data storage, etc.

Conclusion

In conclusion, we have performed several steps in the development of our project. We have defined our requirements, brainstormed possibilities for fulfilling those requirements, analyzed each of the possibilities, and chosen our project design. We then took the design and analyzed it thoroughly. And finally, we implemented our design to fulfill the project requirements.

References

1. ADE7757A | Single Phase Energy Metering IC with Integrated Oscillator and Reverse Power Indication | Energy Measurement | Analog to Digital Converters | Analog Devices. 3 May 2011 <http://www.analog.com/en/analog-to-digital-converters/energy-measurement/ade7757a/products/product.html>.

2. Allied Electronics – Electronic Parts and Components Distributor. 30 April 2011 <http://www.alliedelec.com/>.

3. C8051F35x. 30 April 2011 <http://www.silabs.com/products/mcu/mixed-signalmcu/Pages/C8051F35x.aspx>.

4. CEL. 3 May 2011 <http://www.cel.com/parts.do?command=load&idRootPart=121>.

5. Electronic Components Distributor | DigiKey Corp. | US Home Page. 30 April 2011 <http://www.digikey.com/>.

6. P3 - Kill A Watt. 3 May 2011 <http://www.p3international.com/products/special/P4400/P4400-CE.html>.

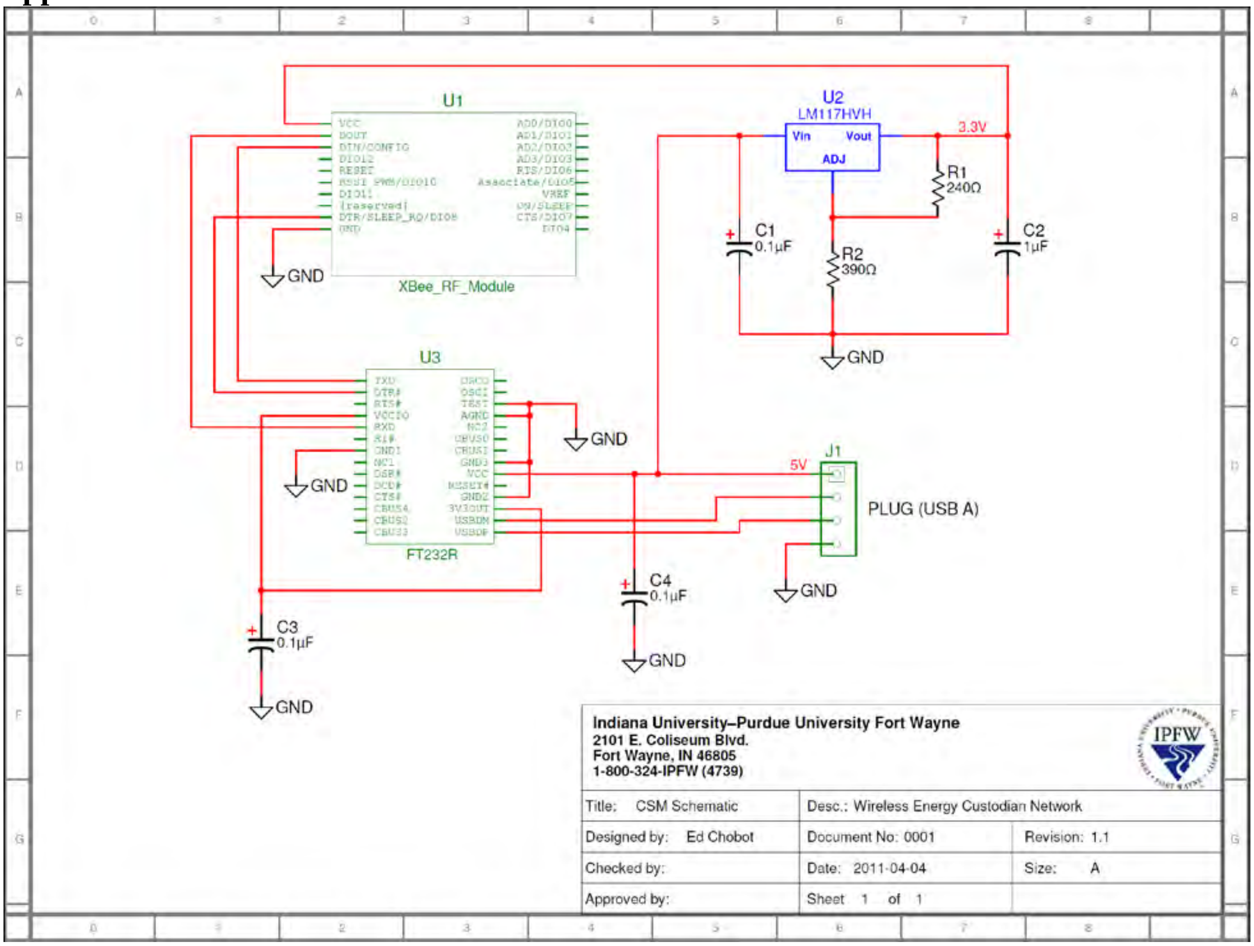
7. Student Program. 30 April 2011 <http://www.4pcb.com/index.php?load=content&page_id=134>.

8. TED 1000 | In-home Electricity Monitor. 3 May 2011 <http://www.theenergydetective.com/store/ted-1000>.

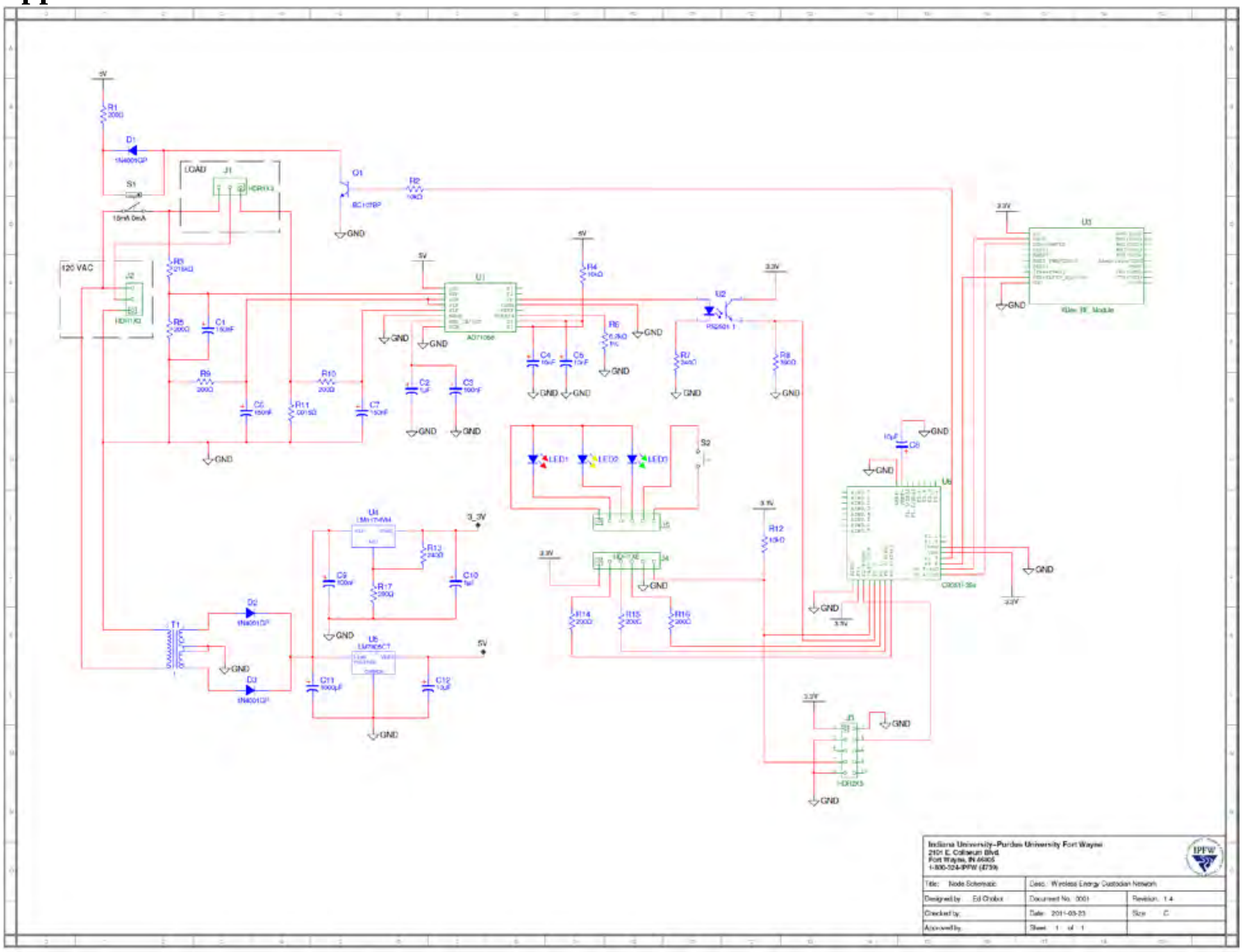
9. USB Debug Adapter. 30 April 2011 <http://www.silabs.com/products/mcu/Pages/USBDebug.aspx>.

10. XBee Explorer Dongle - SparkFun Electronics. 30 April 2011 <http://www.sparkfun.com/products/9819>.

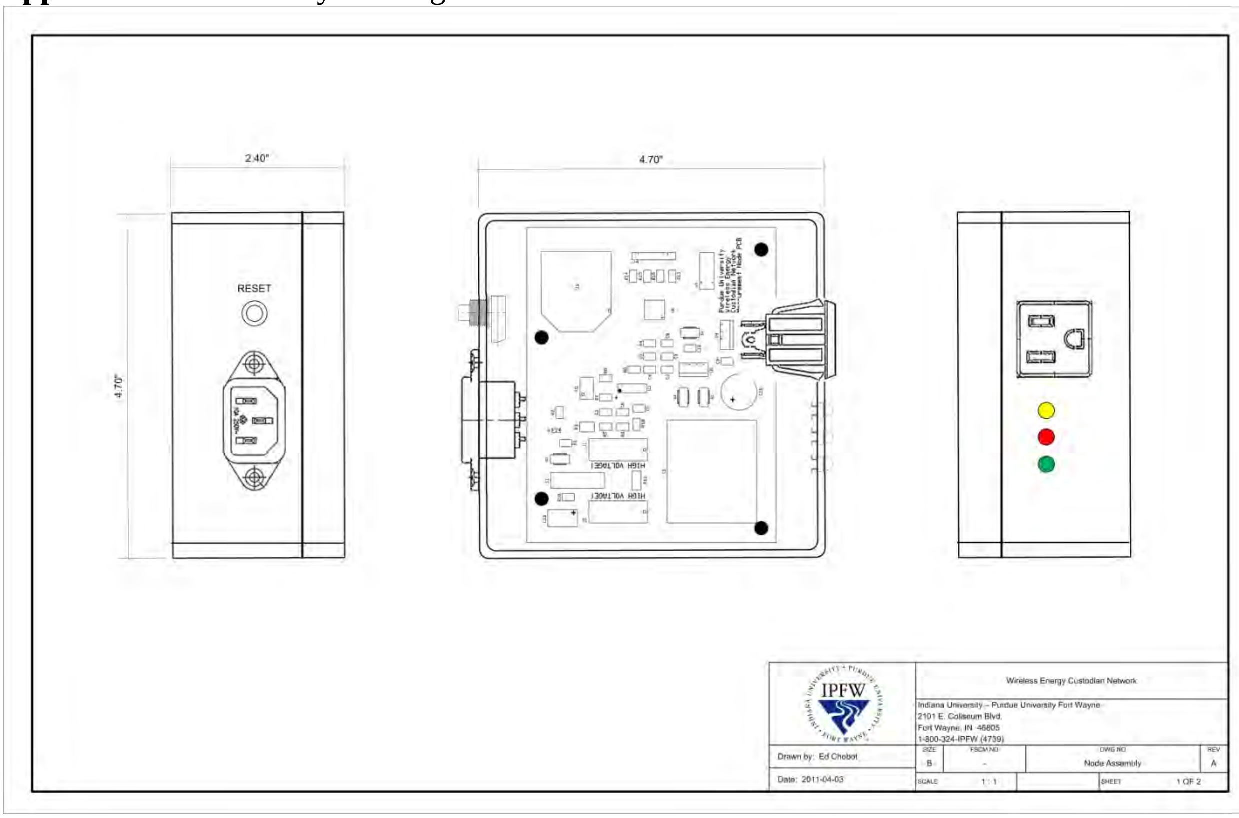
Appendix I: Schematic for Central Server Module.



Appendix II: Schematic for Measurement Node.



Appendix III: Assembly drawing for Measurement Node.



Appendix IV: Bill of materials for the Central Server Module.

SUPPLIER	MANUFACTURER	PART NUMBER	PACKAGE	DESCRIPTION	QUAN-TITY	NOMINAL VALUE	TOLER-ANCE	REF DES
ADVANDED CIRCUITS	ADVANDED CIRCUITS			PRINTED CIRCUIT BOARD	1	N/A	N/A	
ALLIED ELECTRONICS	HAMMOND MANUFACTURING			ENCLOSURE	1	N/A	N/A	
DIGIKEY	AVX CORPORATION	12065C104MAT2A	1206 (3216 Metric)	CAPACITOR, CERAMIC	3	0.1 uF, 50 V	20%	C1,C3,C4
DIGIKEY	AVX CORPORATION	TAJA105M035RNJ	1206 (3216 Metric)	CAPACITOR, TANTALUM	1	1 uF, 35 V	20%	C2
DIGIKEY	MOLEX	480372200	Surface Mount, Right Angle, Horizontal	CONN PLUG USB A 4POS SMD R/A	1	N/A	N/A	J1
DIGIKEY	FAIRCHILD SEMICONDUCTOR	LM317T	TO-220-3	IC REG ADJ 1.5A	1	adjust to 3.3 V		U2
DIGIKEY	FTDI, Future Technology Devices International Ltd	FT232RL	28-SSOP	IC USB FS SERIAL UART	1	N/A	N/A	U3
DIGIKEY	PANASONIC - ECG	ERJ-8ENF2400V	1206 (3216 Metric)	RESISTOR - 1/4 W	1	240 Ohm	1%	R1
DIGIKEY	PANASONIC - ECG	ERJ-8ENF3900V	1206 (3216 Metric)	RESISTOR - 1/4 W	1	390 Ohm	1%	R2
DIGIKEY	DIGI INTERNATIONAL	XB24-ACI-001	PCB - THROUGH HOLE	XBEE RF MODULE W/CHIP ANT	1	N/A	N/A	U1

Appendix V: Bill of materials for each Measurement Node.

SUPPLIER	MANUFACTURER	PART NUMBER	PACKAGE	DESCRIPTION	QUAN -TITY	NOMINAL VALUE	TOLER -ANCE	REF DES
ADVANCED CIRCUITS	ADVANCED CIRCUITS			PRINTED CIRCUIT BOARD	1	N/A	N/A	
ALLIED ELECTRONICS	HAMMOND MANUFACTURING	1554NGY		ENCLOSURE; NEMA; ABS PLASTIC; UL94 HB; LT GRAY; 4.7x4.7x2.4in	1	N/A	N/A	
DIGIKEY	EPCOS Inc.	B32620A3104 J	PCB - THROUGH HOLE	CAPACITOR	1	0.1 uF, 140 V	5%	C13
DIGIKEY	PANASONIC - ECG	ECA-1VM102	RADIAL, CAN	CAPACITOR, ALUMINUM	1	1000 uF, 35 V	20%	C11
DIGIKEY	YAGEO	CC1206KRX7 R9BB103	1206 (3216 Metric)	CAPACITOR, CERAMIC	2	10 nF, 50 V	10%	C4,C5
DIGIKEY	AVX CORPORATION	12065C104M AT2A	1206 (3216 Metric)	CAPACITOR, CERAMIC	2	100 nF, 50 V	20%	C3,C9
DIGIKEY	PANASONIC - ECG	ECJ-3VB1C154K	1206 (3216 Metric)	CAPACITOR, CERAMIC	3	150 nF, 16 V	10%	C1,C6,C7
DIGIKEY	AVX CORPORATION	TPSA106M01 0R1800	1206 (3216 Metric)	CAPACITOR, TANTALUM	2	10 uF, 10 V	20%	C8,C12
DIGIKEY	AVX CORPORATION	TAJA105M03 5RNJ	1206 (3216 Metric)	CAPACITOR, TANTALUM	2	1 uF, 35 V	20%	C2,C10
DIGIKEY	TYCO ELECTRONICS	350825-1	PCB - THROUGH HOLE	CONN HEADER 3POS VERT .250 TIN	2	N/A	N/A	J1,J2
DIGIKEY	TYCO ELECTRONICS	350218-1		CONN PIN 20-14AWG TIN UMNL	6	N/A	N/A	
DIGIKEY	TYCO ELECTRONICS	350766-1		CONN PLUG 3POS 94V-0 UNI-MATE	2	N/A	N/A	
DIGIKEY	QUALTEK	312011-01	POWER CORD	CORD 16AWG 3COND M/F BLK 79" SJT	1	125 V, 15 A	N/A	
DIGIKEY	FAIRCHILD SEMICONDUCTOR	S1A	DO-214AC, SMA	DIODE GEN PURPOSE 50V 1A	4	N/A	N/A	D1,D2,D3, D4
DIGIKEY	ANALOG DEVICES	AD71056ARZ	16 SOIC	IC ENERGY METERING 1 PHASE	1	N/A	N/A	U1
DIGIKEY	FAIRCHILD SEMICONDUCTOR	LM317T	TO-220-3	IC REG ADJ 1.5A	1	adjust to 3.3 V		U4
DIGIKEY	FAIRCHILD SEMICONDUCTOR	LM7805CT-ND	TO-220-3	IC REGULATOR - 1 A, 5 V	1	5 V	4%	U5
DIGIKEY	QUALTEK	703W-00/07	PANEL MOUNT, SCREW HOLE	IEC 320-C14 AC RECEPTACLE	1	125 V, 15 A	N/A	
DIGIKEY	TYCO ELECTRONICS	5103308-1	PCB - THROUGH HOLE	JTAG HEADER (2x5 - 0.1" SPACING)	1	N/A	N/A	J3
DIGIKEY	BIVAR INC	492-1317-ND	RADIAL - 2 LEADS	LED (GREEN)	1	N/A	N/A	LED 3
DIGIKEY	BIVAR INC	492-1320-ND	RADIAL - 2 LEADS	LED (RED)	1	N/A	N/A	LED 1
DIGIKEY	BIVAR INC	492-1322-ND	RADIAL - 2 LEADS	LED (YELLOW)	1	N/A	N/A	LED 2
DIGIKEY	TE Connectivity	3-644456-6	PCB - THROUGH HOLE	LED/RESET HEADER (1x6 - 0.1 SPACING)	1	N/A	N/A	J4
DIGIKEY	TE Connectivity	3-644512-6	WIRING CONNECTOR	LED/RESET WIRING CONNECTOR	1	N/A	N/A	J5
DIGIKEY	SILICON LABORATORIES	C8051F350-GQ	32-LQFP	MICROCONTROLLER	1	N/A	N/A	U6
DIGIKEY	NEC	PS2501-1-L-A	4-DIP	OPTOCOUPLER 1CH TRANS	1	N/A	N/A	U2
DIGIKEY	QUALTEK	738W-X2/03	PANEL MOUNT, SNAP-IN	OUTLET PWR NEMA 5-15R	1	125 V, 15 A	N/A	
DIGIKEY	TAMURA	3FS-312	PCB - THROUGH HOLE	POWER TRANSFORMER	1	2.4 VA, 0.2 A, 12.6 VCT	N/A	T1
DIGIKEY	JUDCO MANUFACTURING	40-3429-00	PANEL MOUNT	PUSH BUTTON SPST N/O MOM (RESET)	1	N/A	N/A	S2
DIGIKEY	SHARP MICROELECTRONICS	S216S02F	4-SIP	RELAY SSR 240VAC 16A	1	240V, 16A	N/A	S1
DIGIKEY	PANASONIC - ECG	ERJ-M1WTF1M5U	2512 (6432 Metric)	RESISTOR - 1 W	1	0.0015 Ohm	1%	R11
DIGIKEY	PANASONIC - ECG	ERJ-14NF2153U	1210 (3225 Metric)	RESISTOR - 1/2 W	1	215 kOhm	1%	R3
DIGIKEY	PANASONIC - ECG	ERJ-8ENF2400V	1206 (3216 Metric)	RESISTOR - 1/4 W	2	240 Ohm	1%	R7,R13
DIGIKEY	PANASONIC - ECG	ERJ-8ENF3900V	1206 (3216 Metric)	RESISTOR - 1/4 W	2	390 Ohm	1%	R8,R17
DIGIKEY	PANASONIC - ECG	ERJ-8ENF6201V	1206 (3216 Metric)	RESISTOR - 1/4 W	1	6.2 kOhm	1%	R6
DIGIKEY	PANASONIC - ECG	ERJ-8GEYJ201V	1206 (3216 Metric)	RESISTOR - 1/4 W	7	200 Ohm	5%	R1,R5,R9, R10,R14,R15,R16
DIGIKEY	PANASONIC - ECG	ERJ-8GEYJ103V	1206 (3216 Metric)	RESISTOR - 1/4 W	2	10 kOhm	5%	R2,R4,R12
DIGIKEY	PANASONIC - ECG	ERJ-8GEYJ470V	1206 (3216 Metric)	RESISTOR - 1/4 W	1	47 Ohm	5%	R18
DIGIKEY	DIODES INC	BC817-16-7-F	SOT-23-3	TRANSISTOR NPN 45V	1	N/A	N/A	Q1
DIGIKEY	DIGI INTERNATIONAL	XB24-ACI-001	PCB - THROUGH HOLE	XBEE RF MODULE W/CHIP ANT	1	N/A	N/A	U3
SPARKFUN ELECTRONICS	SPARKFUN ELECTRONICS	PRT-08272	PCB - THROUGH HOLE	XBEE SOCKET (1x10 - 2 mm SPACING)	2	N/A	N/A	

Appendix V: Software for the Measurement Node.

```
// Program setup for Node 1
#include <reg51.h>

// Special function register addresses
sfr XBR1=0xE2;
sfr XBR0=0xE1;
sfr P0MDIN=0xF1;
sfr PCA0MD=0xD9;
sfr P0MDOUT = 0xA4;
sfr CKCON = 0x8E;
sfr OSCICN = 0xB2;

// Output pin signals
sbitgreenLED = P0^1;
sbityellowLED = P0^2;
sbitredLED = P0^3;
sbitSleepBit = P0^6;
sbitControlBit = P0^7;

// Function prototypes
void Transmit(unsigned char UB, unsigned char LB);
void Receive(void);
void Delay(unsigned int x);

void main()
{
        // Initialize variables
        unsigned char LowerByte = 0x00;
        unsigned char UpperByte = 0x00;

        // Initialize registers
        CKCON = 0x00;           // System clock divided by 12
        OSCICN = 0xC2;          // SYSCLK derived from Internal Oscillator divided by 2 (12.25 MHz CPU)
        PCA0MD=0x00;            // Watchdog timer disabled

        P0MDIN =0xFF;           // Set P0 to Digital Inputs
        P0MDOUT = 0xC0;         // Set P0.6 and P0.7 to PUSH-PULL

        IE = 0x90;              // Enable UART0 interrupt
        SCON = 0x50;            // Serial Mode 1, UART0 reception enabled
        XBR0=0x01;              // Enable UART0
        XBR1 =0xD0;             // Weak Pull-ups disabled. Crossbar enabled. T0 routed to Port pin.

        TMOD = 0x25;            // Timer1 set as timer (8-bit mode), Timer0 set as counter (16-bit mode)
        TH1 = 0xCC;             // Set baud rate to 9600 bps (t = 104 us)
        TF1=TF0=0;              // Clear timer flags
        TR1 = 1;                 // Start Timer 1

        // Test LEDs
        greenLED = 0;
        yellowLED = 0;
        redLED = 0;
        Delay(1000);
        greenLED = 1;
        yellowLED = 1;
        redLED = 1;
        Delay(1000);

        // Turn appliance and transceiver ON
        ControlBit = 1;
        greenLED = 0;
        SleepBit = 0;
```

AppendixVI: Software for the Measurement Node (continued)

```
        while(1)
        {
                yellowLED = 0;

                // Clear the counter and the transmit data
                TH0=TL0=0;
                LowerByte = 0x00;
                UpperByte = 0x00;

                // Count from energy metering chip
                TR0=1;                              // Start energy counter
                //Delay(233);                       // Node 0: Wait 357/1.478 = 242 ms
                Delay(241);                         // Node 1: Wait 357/1.423 = 251 ms
                TR0=0;                              // Stop energy counter

                // Clear single pulse
                if(TH0 == 0x00 && TL0 == 0x01)
                {
                        TL0 = 0x00;
                }

                // Prepare counter data for transmission
                LowerByte = 0x1F & TL0;             // Move first 5 bits of TL0 into LowerByte
                LowerByte = LowerByte + 0x40;       // LowerByte, Node 0 (0x00), Node 1 (0x40)
                UpperByte = 0x03&TH0;               // Move the 2 bits of TH0 into UpperByte
                UpperByte = UpperByte<< 3; // Shift D9 and D8 to the left 3 positions
                TL0 = TL0 >> 5;                     // Shift the lower 5 bits out to the right
                UpperByte = UpperByte + TL0;        // Combine D9 and D8 with D7, D6, and D5
                UpperByte = UpperByte + 0x60;       // UpperByte, Node 0 (0x20), Node 1 (0x60)

                // Transmit
                Transmit(UpperByte, LowerByte);     // Transmit data to CSM
                yellowLED = 1;

                // Wait 5 seconds
                Delay(4800);
        }
}

void Transmit(unsigned char FirstByte, unsigned char SecondByte)
{
        IE = 0x80;                          // Disable UART0 interrupt
        TI = 0;                             // Clear transfer interrupt flag
        SBUF = FirstByte;                   // Move first byte into serial buffer register for transmission
        while(TI == 0);                     // Wait for transmission to complete.
        TI = 0;
        SBUF = SecondByte;                  // Move second byte into serial buffer register for transmission
        while(TI == 0);                     // Wait for transmission to complete.
        TI = 0;
        IE = 0x90;                          // Enable UART0 interrupt
}
```

Appendix VII: Software for the Measurement Node (continued)

```
void Receive(void) interrupt 4
{
        if(RI == 1)         // Receive interrupt
        {
                unsigned char ControlByte = SBUF;       // Move received byte from serial buffer register into ControlByte
                RI = 0;                                 // Clear transfer interrupt flag

                // Turn appliance ON, Node 0 (0x81), Node 1 (0x82)
                if (ControlByte == 0x82)
                {
                        ControlBit = 1;
                        greenLED = 0;
                        redLED= 1;
                }

                // Turn appliance OFF, Node 0 (0x83), Node 1 (0x84)
                if (ControlByte == 0x84)
                {
                        ControlBit = 0;
                        greenLED = 1;
                        redLED= 0;
                }
        }
}

void Delay(unsigned int x)
{
        // (1/12 MHz)*12 clock cycles = 1 us
        // Baud Rate: 1/9600 = 104 us
        // (104 us)/(1 us)/2 = 52 = 34h
        // Set Baud Rate: 100h - 34h = CCh
        // 52*(1 us)*20 = 1.04 ms
        // When x = 1, delay is 1.04 ms.
        // When x = 343, delay is 357 ms.
        // When x = 2885, delay is 3 s.
        // When x = 9615, delay is 10 s.
        // When x = 65535, delay is 68 s. (maximum delay)
        unsignedint i;
        unsignedint j;
        for(i=0;i<x;i++)
        {
                for(j=0;j<20;j++)           // (52 us)*20 = 1.04 ms
                {

                        TF1 = 0;            // Clear Timer 1 flag
                        while(TF1 == 0);    // Monitor Timer 1 flag until it rolls over
                        TF1 = 0;            // Clear Timer 1 flag
                }
        }
}
```